AD-A115 480 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 17/5
AUTOMATIC TARGET CUEING IN IR IMAGERY.(U)
DEC 81 N A HAMADANI
UNCLASSIFIED AFIT/GEO/EE/81D-3 NL

1 of 1
AD
A115480

END
DATE
FILMED
7 82
DTIC

AD A115480

DDC

UNITED STATES AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DTIC ELECTE

AFIT/GEO/EE/81D-3

AUTOMATIC TARGET CUEING

IN IR IMAGERY

THESIS

Naser A. Hamadani
AFIT/GEO/EE/81D-3 Flt Lt PAF

AFIT/GEO/EE/81D-3

AUTOMATIC TARGET CUEING
IN IR IMAGERY

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by
Naser A. Hamadani
Flt Lt PAF
Graduate Electro-Optics
December 1981

Approved for Public Release; Distribution Unlimited

Acknowledgements

Praise be to God, without Whose grace nothing is possible. I am deeply indebted to Dr. Matthew Kabrisky, Professor of Electrical Engineering, for his advice and encouragement throughout the study. I also have benefitted greatly from discussions with Lt Col R.J. Carpinella. Special acknowledgement is due to my wife, whose support and prayers were a great incentive throughout my stay at AFIT.

Last, but not the least, I wish to thank Sharon A. Gabriel who typed this thesis.

Naser A. Hamadani

Contents

Page

Acknowledgements ---------- ii

List of Figures ---------- iv

Abstract ---------- v

I. Introduction ---------- 1

Importance ---------- 1
Background ---------- 3
Purpose ---------- 5
Scope ---------- 5

II. Preprocessing ---------- 7

Introduction ---------- 7
Image Enhancement ---------- 9
Edge Detection ---------- 13
Thresholding ---------- 14

III. Target Extraction ---------- 18

Introduction ---------- 18
Closedness ---------- 18
Thinning ---------- 23
Size Criterion ---------- 29

IV. Results and Discussion ---------- 34

Introduction ---------- 34
Results ---------- 34
Discussion ---------- 37

V. Conclusions and Recommendations ---------- 42

Conclusions ---------- 42
Recommendations ---------- 43

Bibliography ---------- 46

APPENDIX A: Data ---------- 47

APPENDIX B: Gray Scale ---------- 51

APPENDIX C: Segmentation ---------- 54

APPENDIX D: Program Listing ---------- 65

Vita ---------- 66

List of Figures

Figure		Page
2-1(a)	A Matrix--	8
2-1(b)	Gray Scale Representation of a Matrix--------	8
2-2	Numbering of Elements in Kirsch Operators----	10
2-3	An Original IR Picture----------------------	11
2-4	Image Enhanced Picture----------------------	12
2-5	Kirsch Edge Matrix--------------------------	15
2-6	Thresholded Picture-------------------------	17
3-1	Results of an Ideal Edge Detector-----------	19
3-3	An Unclosed Cluster-------------------------	20
3-4	Results of Closedness Test------------------	22
3-5	Disadvantage of Closedness Test-------------	21
3-6	Flow Chart of Thinning Algorithm------------	25
3-7	Results of Thinning-------------------------	27
3-8	Results of Thinning-------------------------	28
3-9	Results of Size Test------------------------	30
3-10	Special Case--------------------------------	31
3-11	Final Processed Picture---------------------	33
5-1	Pixel Classification------------------------	45
A-1	Labelling of Record 1 in Data---------------	49
A-2	Labelling of Record 2 in Data---------------	50
B-1	Gray Scale Printing Code--------------------	53
C-1	Picture After Closedness Test---------------	56
C-2	Picture After 1st Step of Segmentation------	57
C-3	Picture After Segmentation------------------	60
C-4	1st Step of Segmentation--------------------	61
C-5	2nd Step of Segmentation--------------------	63

Abstract

This thesis presents an algorithm for detecting man-made objects embedded in low resolution imagery. A modified Kirsch edge operator is used for initial image enhancing. A normal Kirsch operator is then used for edge detection. A two-dimensional threshold for edges and the original intensity detects the pixels on the edges of the objects only. These pixels are then subjected to connectedness and size tests to detect the blobs which most probably represent man-made objects. The algorithm was tried on 325 pictures and a detection probability of 83.3% was achieved. False alarm probability was less than 10%.

I. INTRODUCTION

Importance

Military applications for locating man-made objects in low resolution imagery are commonly referred to as "target cueing" (Ref 1).

Recent advances in image sensor technology have increased the sensitivity and field of view of imaging systems significantly. This is particularly true in the infrared wavelength region with the development of high quality infrared sensors. Because of this, the amount of information presented to the operator of such imaging systems has increased manyfold over previous systems and the operator is burdened with a high task load in attempting to analyze scene content. Tasks involving the detection and recognition of man-made objects (targets) become inefficient and inaccurate as the time available for this task becomes short or as the environment surrounding this task load becomes complicated.

Automatic machine detection and recognition of man-made objects is now viable with recent advances in electronic technology. Some of these include high-density digital memories, CCD analog signal processing components, advances in pattern recognition techniques, and the availability of high quality IR sensors. The area of automatic FLIR target cueing has received much attention recently. Results indicate

that the performance achievable with automatic target cuers can be better than human performance under similar time and environmental constraints.

The problem of pattern classification, as classically defined, aims at analytical techniques by which a given object can be correctly assigned into one of several pre-specified categories. Various approaches have been under active investigation for many years, and several representative results can be found in readily available literature (Refs 2-4). All pattern classification techniques developed to date are predicated for optimal performance upon the availability as input to the classifier of a "well-defined test pattern clearly devoid of any extraneous components irrelevant to the classification task" (Ref 12).

A "well-defined pattern" implies a high resolution image containing all the textural features of the target. With the IR sensors available today, the required resolution is achievable only at close ranges from the targets. In a real combat situation, this closeness may be fatal. From a tactically advantageous distance, the resolution is not high enough to present a "well-defined test pattern" to a classifier.

Therefore, within the context of "real" imagery, a natural question arises as to how one can localize a candidate object in a given image and, following such an operation, how does one extract it from its surrounding background so that as ideal a pattern as possible is presented to the classifier

designed to yield its identity. Implicit in this statement is the partitioning of pattern classification task into three phases. First, a scene is scanned for localization of possible candidate targets; second, each object is extracted from its immediate environment; and third, the raw or feature summarized object is preprocessed through the classifier for final identification.

This thesis report deals with the first two phases of the pattern classification problem, as described above.

Background

FLIR sensors detect radiation of wavelengths in 3-5 or 8-14 micrometer atmospheric windows and derive their images from the variations in their field of view of the IR radiation received. These variations can be due either to variations in the emittance of the scene or to variations in the radiation reflected from the scene. The radiation reflected from the scene can be caused by a variety of natural sources such as clouds, sky, and background. But the radiation reflected is usually less than that associated with a target at ambient temperature. The primary scene signal results from variations in emittance and may be due to variations in temperature or emissivity. That is, the scene is the source generating most of the radiation itself because its inherent temperature differs.

The process of pattern recognition usually involves a sequence of operations including preprocessing, feature

extraction and classification. Preprocessing operations are sometimes applied to correct for known geometric distortions, to filter out redundant data to enhance information contents of the scene, and to transform the data to make it easier to extract features. These features are used to further discard false alarms, if any, and classify the objects if target classification is desired.

The derivation of a useful set of features is the most critical part of the solution of pattern recognition problems. Like visible scenes, thermal scenes also have infinite variety. The choice of features for a given problem, at present, is largely ad hoc. Feature definition, therefore, is strongly dependent upon the application and the kind of preprocessing operations performed. Satisfactory solutions to difficult problems seem to require considerable study and experimentation with large amounts of representative data and a thorough understanding of the underlying physical phenomena which dictate the empirically observable class variations. Moreover, the tools of the analyst often determine the ease with which a solution is obtained and probably the nature of the solution as well.

The essence of the problem is that, at present, there is no mathematical model or basis which could usefully represent the infinite variety of thermal scenes in nature, as a whole or in parts. Therefore, it is the intuition and the judgment of the individual, rather than any underlying theory, which determines the operations performed and features analyzed for picture processing.

Purpose

The purpose of this study was to devise a combination and sequence of operations for a specific set of IR pictures so as to achieve the best possible results. The merit of the results was based on two criteria; maximum detection probability and minimum false alarm probability. Maximization of detection probability was given preference over minimization of false alarm probability. The motivation for this preference was twofold. First, the results of a target cuer are not used to make the final decision; rather, they are an aid to the final decision-making process. Therefore, it is more important to detect all the targets than to detect only the targets. Secondly, the false alarm probability can be minimized to a great extent if the ratio of the actual target size to that of the pictured target is known. This ratio, besides other factors, is dependent upon the characteristics of the sensor. Access to the sensor and its characteristics was not provided because of security reasons.

Scope

The algorithms in this thesis were only tested on data provided by Eglin Air Force Base, Florida. Therefore, the whole process was structured to provide the best possible results for that data. The performance of the algorithms on any other data is not known.

The computation time required by the process is an important operational consideration. Not only accuracy, but accuracy with speed is required in actual hardware. Computer programs for this thesis, however, were not optimized with regard to running time or number of operations required because of academic time limitations. The nature of the calculations used by the algorithm do not seem to suggest any difficulty in practical implementation, and it is anticipated that a dedicated chip implementation should be capable of running in real time.

II. PREPROCESSING

Introduction

In digital picture processing, a picture is represented by a matrix of certain dimensions. Each element of the matrix, shown as a point on the picture, represents a certain area in the actual scene. The value of each element of the matrix is a measure of the temperature of the corresponding area in the actual scene. Figure 2.1(a) shows a 10 x 10 matrix, and Figure 2.1(b) is a gray scale representation of the same matrix.

Before any decision is made as to whether a target is present or not, it is important to localize the areas in the picture which look like potential targets. This process of localizing potential targets is called preprocessing. A point is considered to be a part of a potential target if

i) its intensity is above a certain threshold, and

ii) its contrast with its surrounding points is above a certain threshold.

This chapter gives the details of those operations which select only those points out of the picture which satisfy the above criteria.

Image Enhancement

The underlying assumption of the whole process is that the objects or the targets in the picture appear as a cluster of points, which are above the intensity and contrast threshold. But in real imagery, it does not happen this way. There are many points within the object area which fall below one of the thresholds. This phenomenon affects the picture processing adversely in two ways. First, when looking for targets as small as 2 x 2 or 3 x 3, disappearance of even two or three points from the target area tends to destroy the whole target. Secondly, a large piece of terrain or some other natural object in the scene as hot as the targets tends to appear divided into parts with each part having the appearance of a target. Therefore, it is essential to enhance the quality of image in the sense that low intensity points in an area of high intensity should be brought to the same intensity level.

In the process of edge detection, many edge operators available in literature were tried. One of the effective and interesting edge operators tried was the Kirsch operator (Ref 5). In the Kirsch operator, every point, p, in the picture is given an edge value according to the contrast function

$$\max \left[1, \max_{i=0}^{7} \left| 5(a_i+a_{i+1}+a_{i+2}) - 3(a_{i+3}+a_{i+4} \cdots +a_{i+7}) \right| \right]$$

where all subscripts are evaluated modulo 8.

a_0	a_1	a_2
a_7	P	a_3
a_6	a_5	a_4

Figure 2.2. Numbering of Elements in the Neighborhood of an Arbitrary Point, P, in Kirsch Operator

To study the effect of coefficients 5 and 3 on the performance of the operator, each coefficient was varied from 1 to 10, keeping the other constant. It was observed that the contrast function

$$\max\left[1, \max_{i=0}^{7}\left|10(a_i + a_{i+1} + a_{i+2}) - (a_{i+3} + a_{i+4} + \cdots a_{i+8})\right|\right]$$

behaved as an image enhancing function. Figure 2.4 is an image_enhanced version of Figure 2.3.

The image enhancing function, also, increases the size of the targets. But that need not present any problem. The purpose is to detect a target and the analysis of the results shows the bigger the target, the better the detection. So the effect of image enhancing on the ultimate detection is especially beneficial.

Figure 2-3. An Original IR Picture

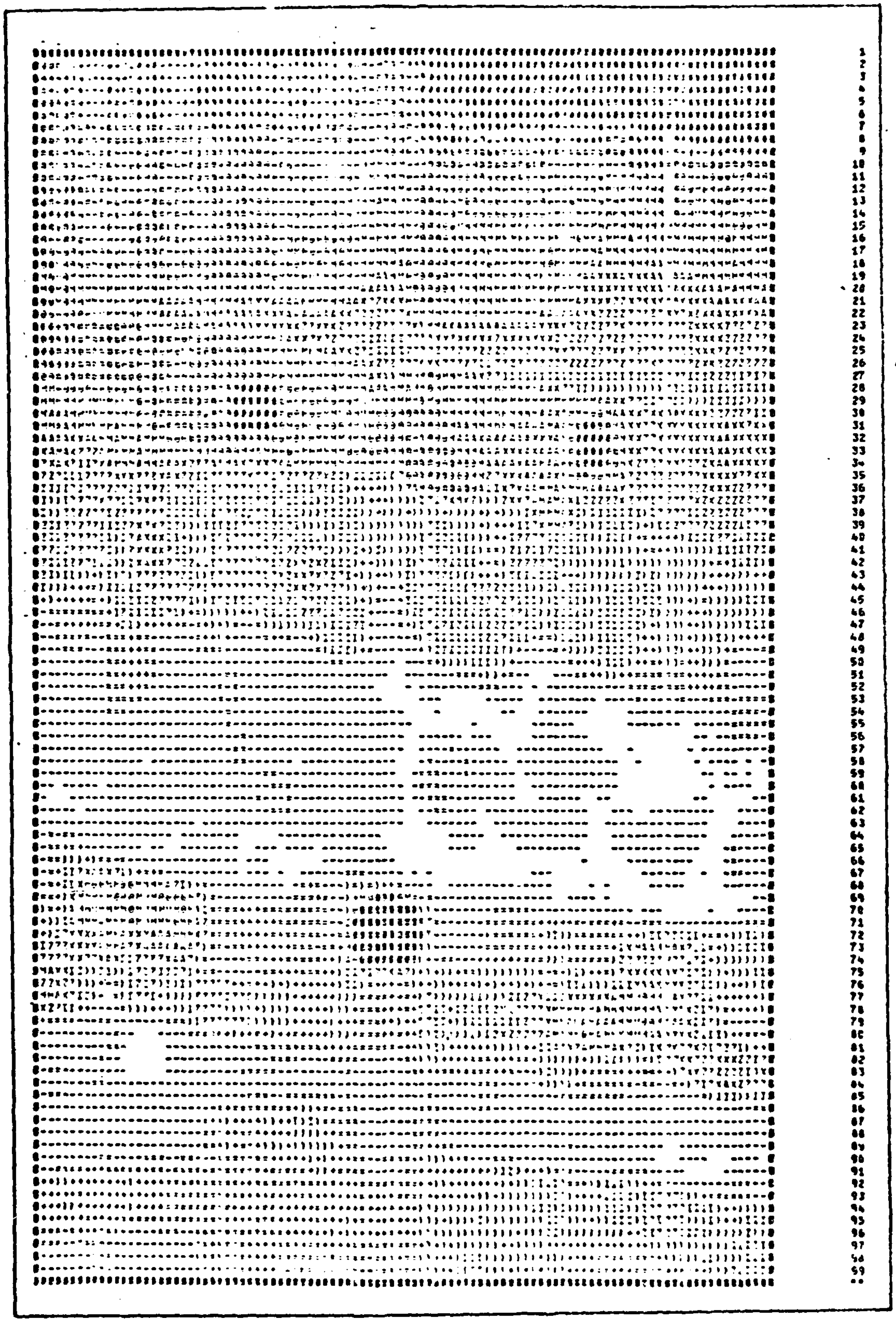


Figure 2-4. Image Enhanced Picture

Edge Detection

Image edges can be defined as local changes or discontinuities in an image attribute such as luminance or texture (Ref 6). These changes are important in the analysis of images because they often provide an indication of the physical extent of the objects within the image. In target cueing, the significance of edges becomes more important because, in the absence of textural information in the images, edges are the only feature which can be readily extracted.

There are many techniques which can be used in edge detection. These include simple differential operators, template matching, least square edge fitting, and techniques based on statistical detection theory. There are also many heuristic methods developed for edge detection. Surveys of edge detection techniques are readily available in the literature (Refs 7-10).

For the purpose of this thesis, the Robert operator (Ref 11), Sobel Operator (Ref 13), and Kirsch operator (Ref 5) were experimented with. The Kirsch operator was found to be the best as far as results were concerned. This choice was based upon the empirical results obtained from processing of the pictures from the given data.

The Kirsch operator assigns an edge value to each point in the picture by calculating a contrast function,

$$\max\left[\left|5(a_i + a_{i+1} + a_{i+2}) - 3(a_{i+3} + a_{i+4} + \cdots a_{i+8})\right|\right]$$

where all subscripts are evaluated modulo 8. Figure 2.2 gives the details of the numbering of elements in the neighborhood of an arbitrary point, P. This non-isotropic contrast function is related to the magnitude of the gradient of the original brightness function. It is non-symmetric and sensitive to small changes in the value of the gradient. Figure 2-5 is the Kirsch operated version of Figure 2-3 after it had been image enhanced (Figure 2-4).

The Kirsch operator was developed for the processing of biological images where the objects to be detected are small biological cells. This is exactly the problem in target cueing. The targets to be detected are at large distances and therefore very small. That seems to explain the better performance of Kirsch operators.

Thresholding

In this thesis, a target is defined as a cluster of points having intensity above a certain threshold and edge values above a certain threshold. The problem lies in selecting the threshold so that no target points are lost and no noise points are introduced. It is highly improbable that a universal threshold should exist which will be a criterion of acceptance and rejection of a point for all

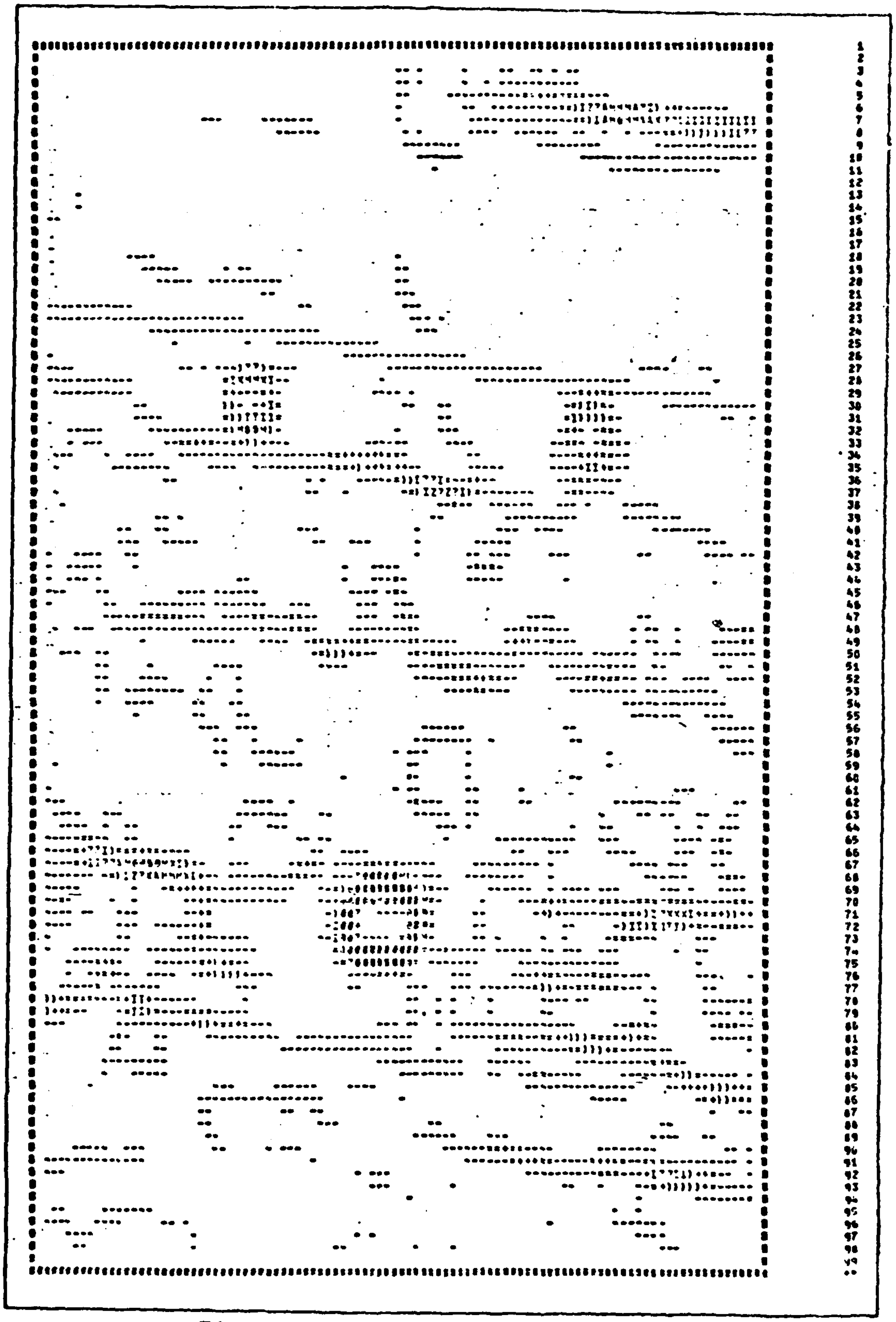


Figure 2-5. Kirsch Edge Matrix

pictures, even in a given data set. Instead, it is more appropriate to dynamically select a threshold for every picture separately. By trial and error, a threshold of mean plus variance of a picture was found to be a good threshold.

The process calculates the mean and variance of the original intensity function. The edge values for each point are calculated using Kirsch contrast function and the mean and variance of these edge values are calculated. Now each point in the picture has two values associated with it, the intensity value and the edge value. At this stage, each point is tested for a two dimensional threshold, one each for the two values associated with it. A point is retained as a constituent of a target if its intensity value is greater than mean plus variance of picture intensity and its edge value is greater than mean plus variance of picture edge values.

Figure 2.6 is a 2-D thresholded version of Figure 2.7.

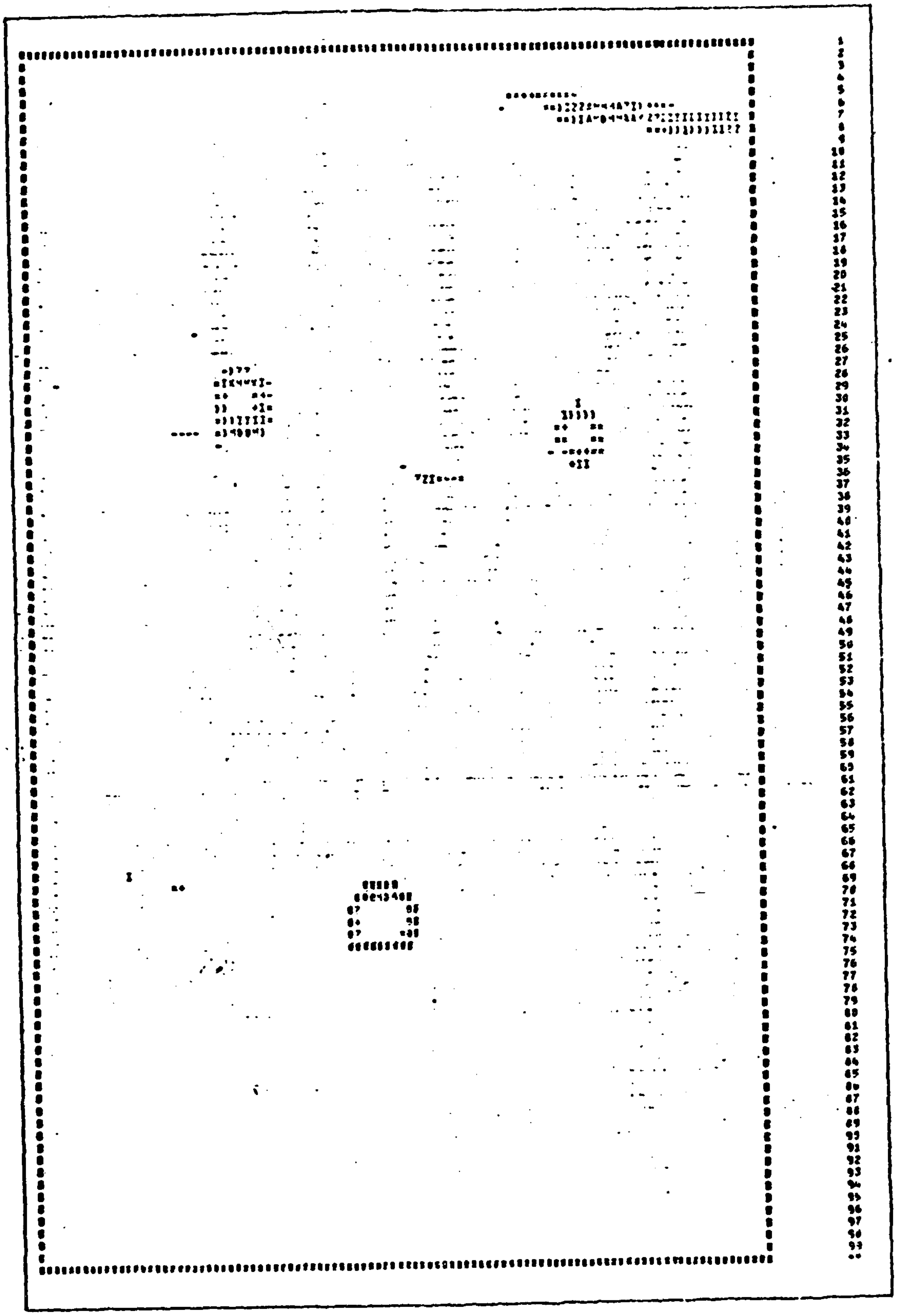


Figure 2-6. Thresholded Picture

III. Target Extraction

Introduction

The preprocessing operations described in Chapter II extract those points from the picture which have high intensity and edge levels. These points, being constituents of a target, appear in the form of clusters. These clusters are either the potential targets or those areas of the background which appear as targets in the imagery. In the absence of textural details inherent in low resolution imagery, some statistical and geometrical criteria have to be defined so that the potential targets and the background noises appearing as targets can be differentiated. These criteria are defined on the bases of observation and analyses of the results obtained after preprocessing in conjunction with a priori knowledge of presence and location of targets provided with the data.

This chapter defines the criteria which yield the best results and describes the operations which have to be performed after the preprocessing.

Closedness - Two-Neighbor Criterion

The preprocessing extracts only those points from the picture which have high intensity and edge levels. High intensity level pixels are those which constitute a target or those background areas which appear as targets. From

among such points, only those will have high edge values which lie on the boundaries of these targets or target-like backgrounds. Ideally, then, only the outlines of these targets should appear in the picture after the preprocessing. Figure 3.1 depicts this situation.

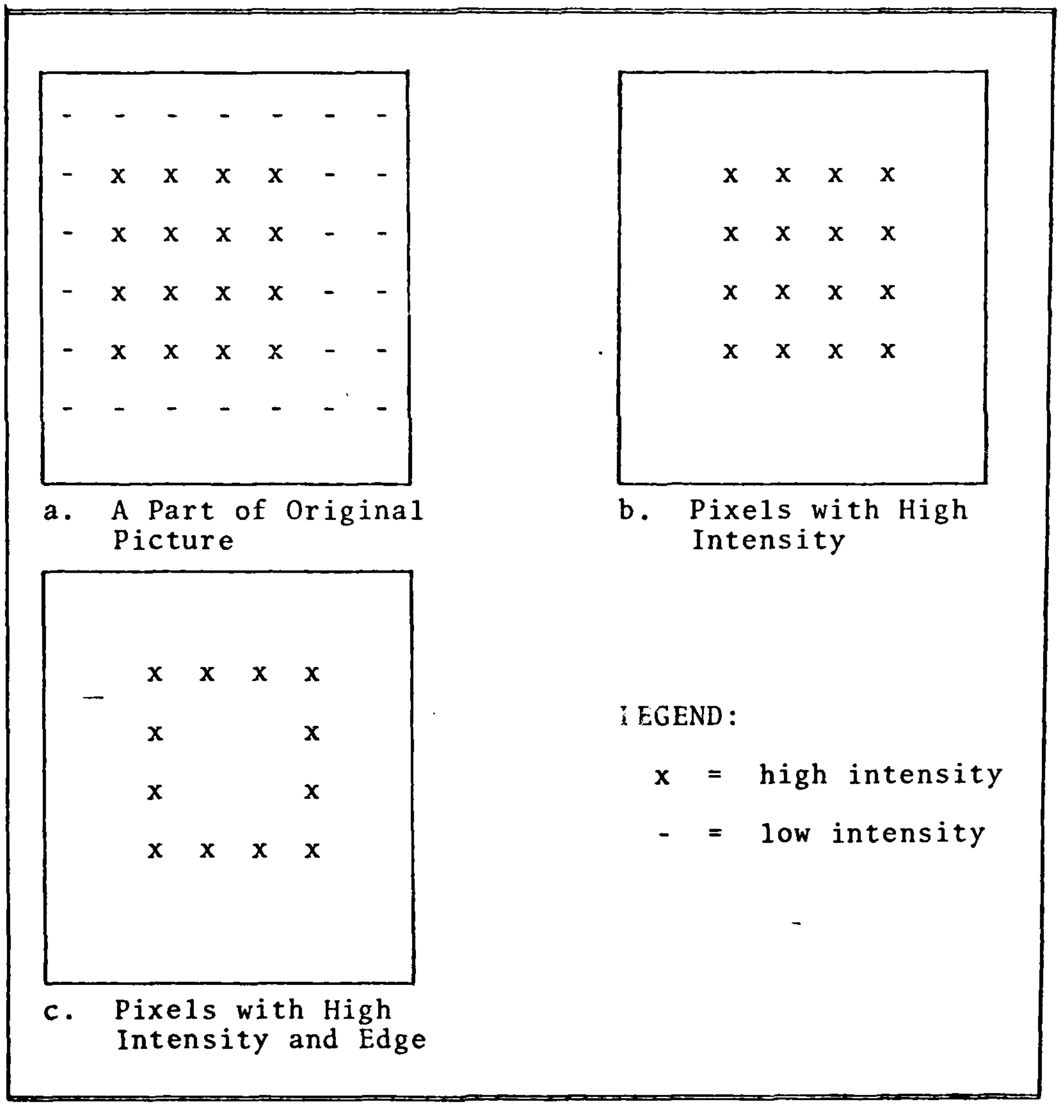
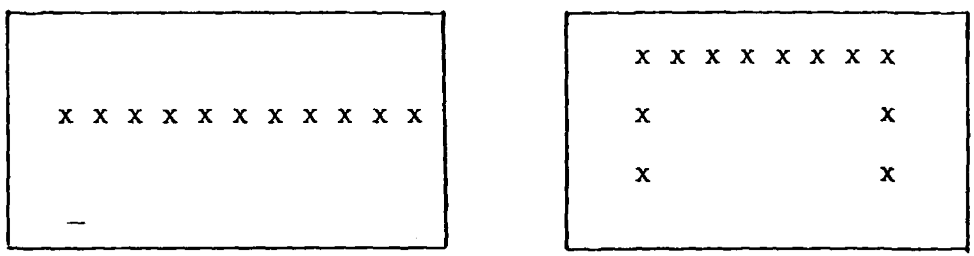


Figure 3.1. Results of an Ideal Edge Detector

But this is not what actually happens. There are many pixels left which are unwanted, as shown in Figure 2.6. Therefore, it is necessary that the picture be cleaned of such points.

The criterion used to get rid of the unwanted pixels was based on the premise that all man-made objects have closed boundaries. If Figure 3.1(c) is observed carefully, it would be noticed that each pixel in the outline of the target has two adjacent neighbors. Moreover, all these points in the outline are normal to each other. Diagonal neighbors do not matter. If, on the other hand, a cluster of points forming a straight line or a curve, as shown in Figure 3.3, is observed, the situation is reversed. The



a. Straight Line B. Curved Cluster

Figure 3.3. An Unclosed Cluster

points on the extremes do not have two neighbors. But another important observation can also be made; namely, that this operation of checking the neighbors has to be

reiterated many times to clean the picture of all the points in a cluster, all of whose constituents do not have two neighbors. Figure 3.4 is the same picture as in Figure 2.6, but after closedness test has been made on it.

There is one disadvantage of this criterion. Figure 3.5 is a representation of a target and a non-ideal result of an edge operator on it. To a human eye, Figure 3.5(b)

```
-  -  -  -  -  -  -
-  x  -  x  x  x  -
-  x  x  x  x  x  -
-  x  x  x  x  x  -
-  x  x  x  x  x  -
-  -  -  -  -  -
```

a. Target

```
x     x  x  x
x  x        x
x           x
x  x  x  x  x
```

b. Result of a Non-Ideal Edge Operator

LEGEND: - = Low Intensity
x = High Intensity

Figure 3.5. A Disadvantage of Closedness Criterion

is a perfectly good target. But not all points constituting the target have two neighbors. Therefore, the target is rejected by the closedness test, simply because one point is missing.

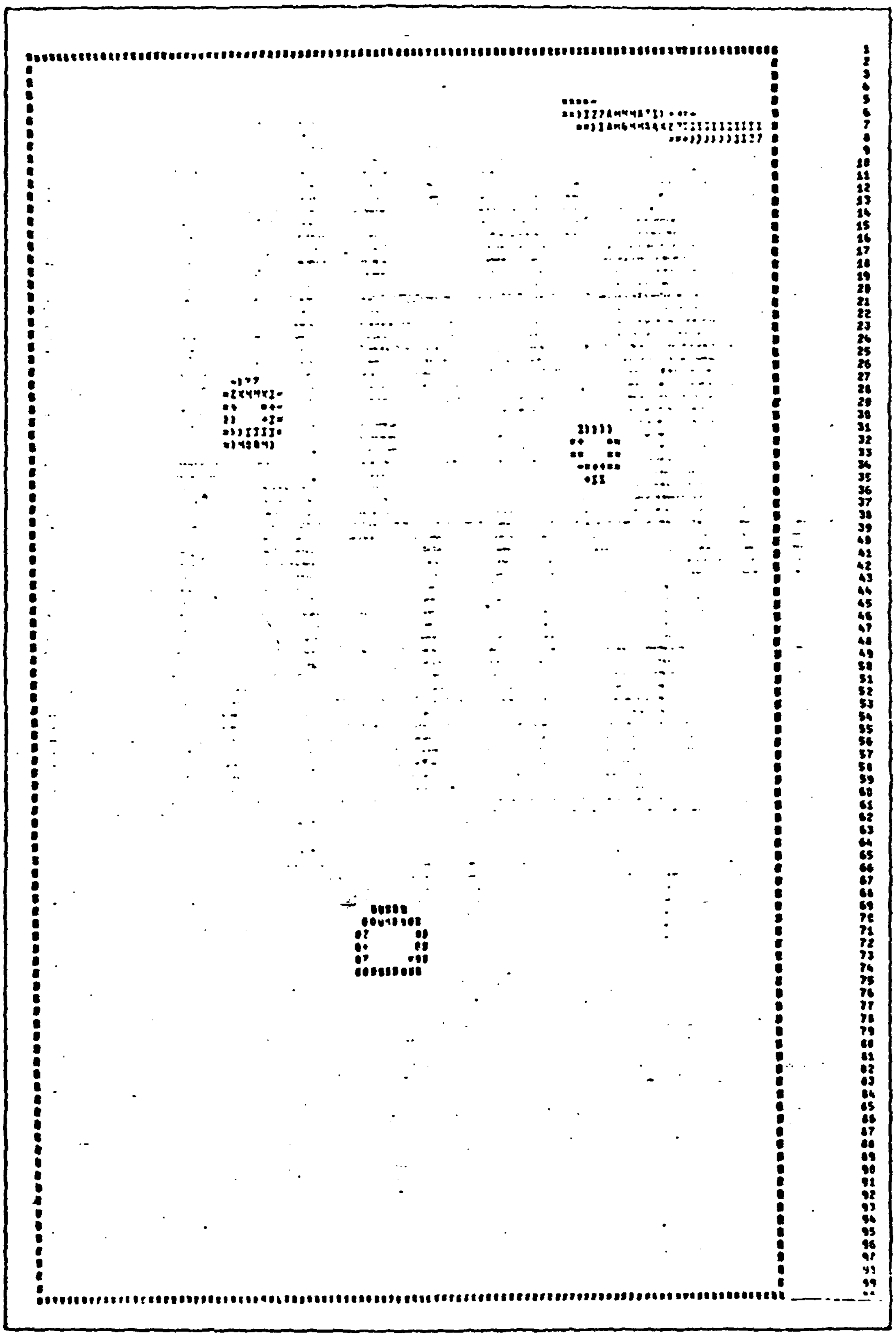


Figure 3-4. Results of Closedness Test

The disadvantage of the closedness criterion, though, does not pose a big problem. The enhancement algorithm, explained in Chapter II, tries to ensure that no such points in a cluster remain below the threshold which would cause the whole cluster to be unstable for closedness criterion.

Another reason why this disadvantage is not of much concern is the non-ideal response of edge detectors, which is explained in the next section.

<u>Thinning - Maximum Rectangle Criterion</u>

Figures 3.1 and 3.5 are the results of a hypothetical ideal edge detector. It is unreal, also. Figure 3.4 are the results of real edge detectors. It is obvious from the two cases that edges do not appear to be fine, thin lines as we desire them to be. They are thicker than they appear to a human eye. It may be because of one or both of the following reasons. First, the basic mathematical model used to represent the edges in the formulation of the edge detector may be defective. Second, the addition of noise, due to different sources, corrupts the edges to an extent that they do not exist in the imagery as fine, thin lines.

Blurring of the edges due to noise and/or the inability of edge detectors to detect them makes it difficult to further analyze the targets as they are. This is due to the fact that the size information of the target is not clear. This is obvious from Figure 3.5. Therefore, it is necessary

to thin the edges to simple lines before they can be further analyzed.

The thinning algorithm used is that of fitting a maximum rectangle within the boundaries defined by earlier operations. The process compares the outermost rows and columns of a target with the respective next-to-outermost row or column. The comparison is made with respect to the number of pixels detected in each row and column. This operation is performed sequentially on all sides of the target. The row or column with maximum detected pixels on each of the four sides is defined as the final representation of the edge on that side. Figure 3.6 presents a flow chart of this whole process.

The underlying idea behind this process is that of probability. The edge detection algorithm detects more than one line in each direction for one actually existing edge line. Each detected line is constituted by a certain number of pixels which fulfill the criterion of an edge existence at that point. It is logical to assume that a line consisting of more pixels is more probable to represent an edge than another line with less pixels. Figure 3.7 illustrates this point.

Figure 3.8 is an actual picture after edge detection and thinning.

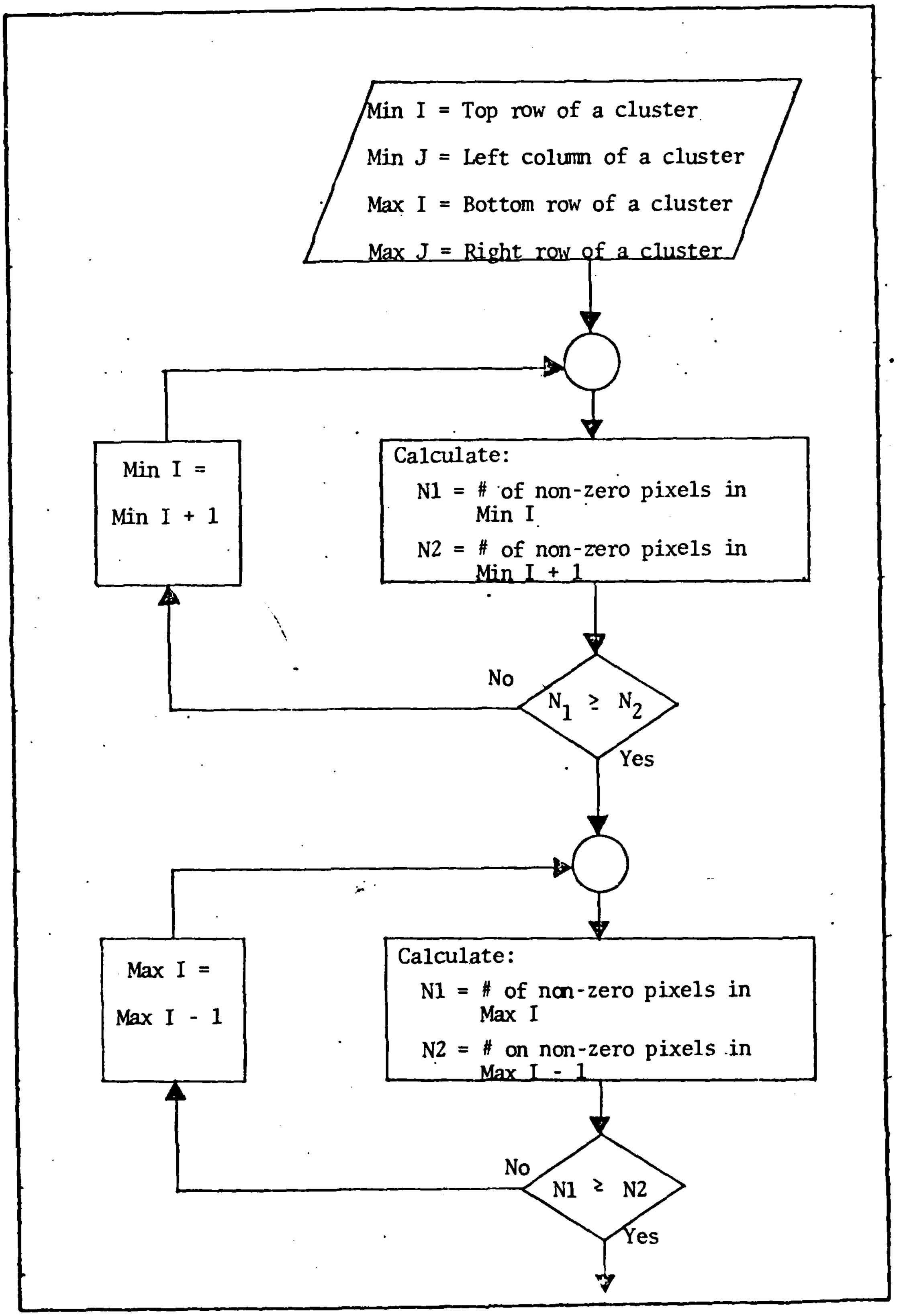
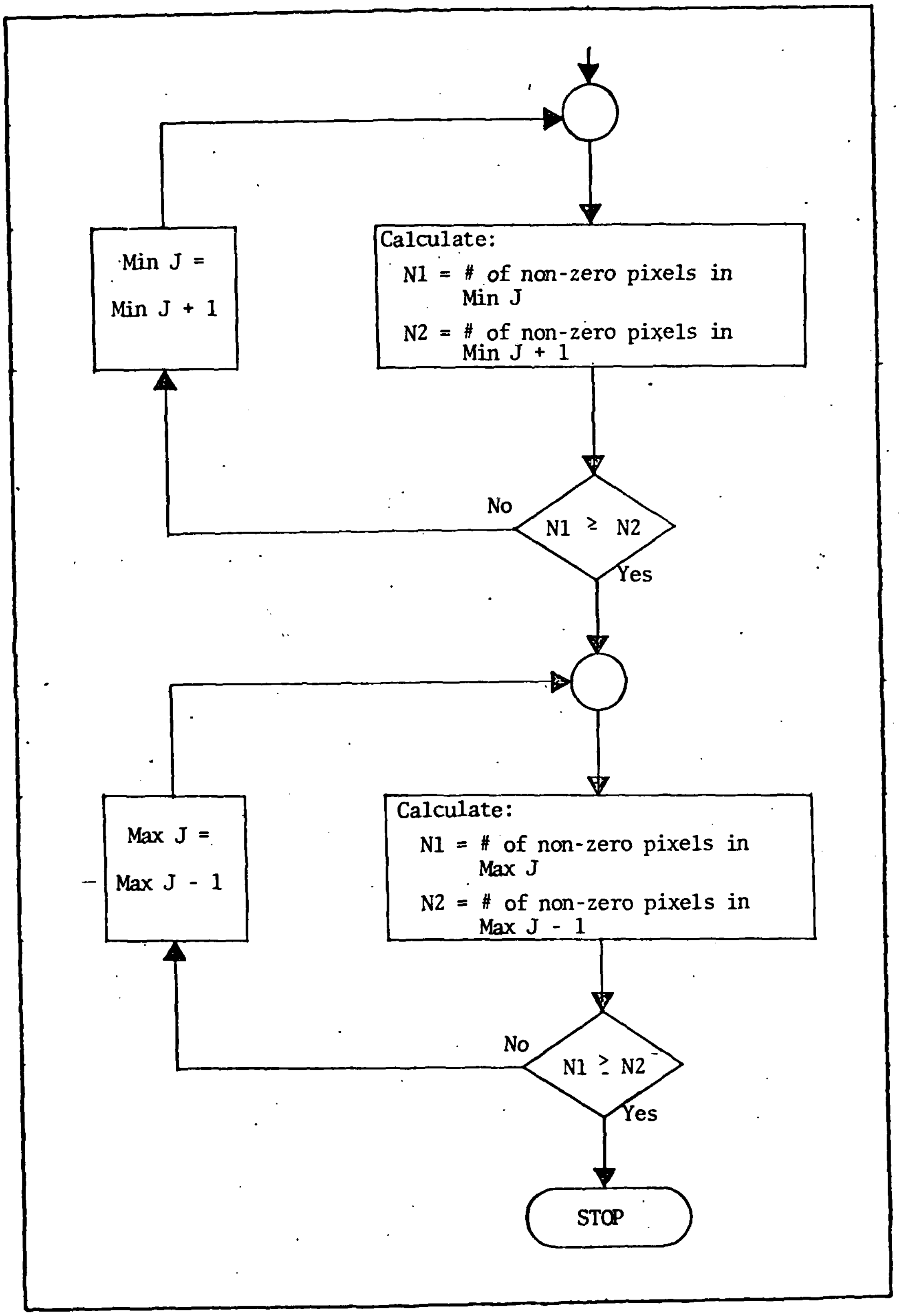


Figure 3-6. Thinning


Calculate:
N1 = # of non-zero pixels in Min J
N2 = # of non-zero pixels in Min J + 1
Min J =
Min J + 1
No
N1 ≥ N2
Yes
Calculate:
N1 = # of non-zero pixels in Max J
N2 = # of non-zero pixels in Max J - 1
Max J =
Max J - 1
No
N1 ≥ N2
Yes
STOP

```
x       x x     x x
x   x x x x x x x x
x   x     x x   x x
    x           x x
x   x           x x
    x           x x
  x x x x     x x x
  x x x x x x x x x
```

a. Edge Detector Output

```
x x x x x x x x
x             x
x             x
x             x
x             x
x             x
x x x x x x x x
```

b. (a) after Thinning

Figure 3-7. Results of Thinning

The problems that might arise out of this algorithm are the cases of target of the sizes 2 x 2 or 3 x 3. In these cases, the opposite edges are either adjacent or connected by false and noisy edges. Such cases are taken care of by image enhancement algorithm explained in Chapter II. That algorithm, in the process of enchancing, also increases the size of the target. This increase in size for a 2 x 2 or 3 x 3 target size is sufficient to overcome the problems posed by thinning algorithm.

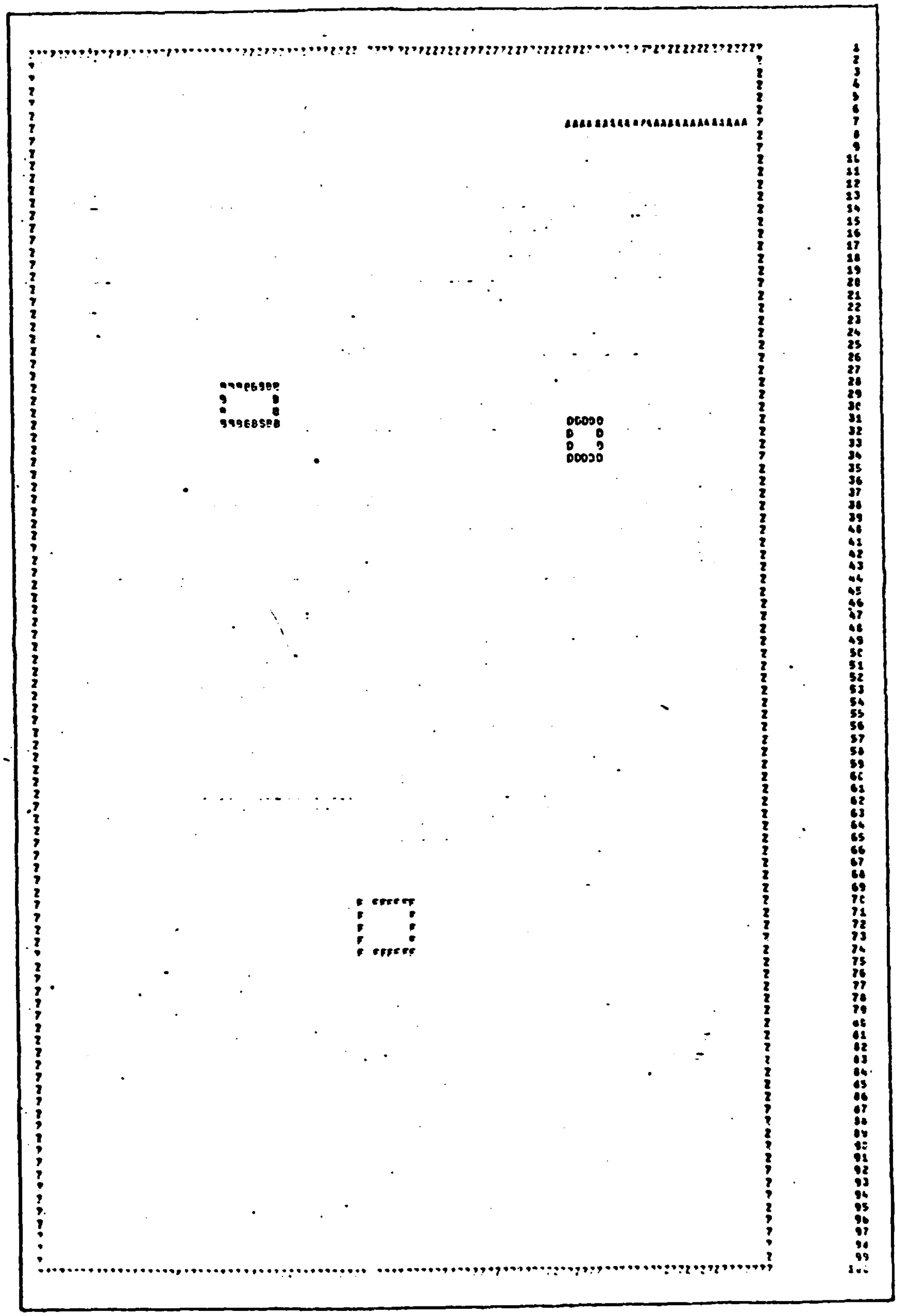


Figure 3-8. After Thinning

Size Criterion

After thinning has been accomplished, the targets appear in a form where they can be checked for their actual physical sizes. Because of image enhancement, the sizes of the targets are increased. That implies a target would appear only larger than 3 x 3 in size. Therefore, all the targets which are smaller than three pixels in any of the dimensions should be discarded. The limit on the other extreme of size is strongly dependent upon sensor optics, range and actual size of the targets. Since all of these parameters are classified and were not provided for this study, it is difficult to set any logical limits on the size. But just by observation of all the pictures in the unclassified data set, a limit of 8:1 was determined. That is, the two dimensions of the target could be in a ratio of 8:1. Any ratio greater than that will make the target too large in one dimension with respect to the other to represent a real man-made object. Figure 3-9 is a final version of Figure 3.8.

Another consideration which becomes very important in certain cases is the one shown in Target A of Figure 3.10. One side of the target is completely missing. What this example indicates is that a certain percentage of pixels in the circumference of the object has to be detected if that object is to be classified as a target. But the question is -- what percentage? The answer to this question again

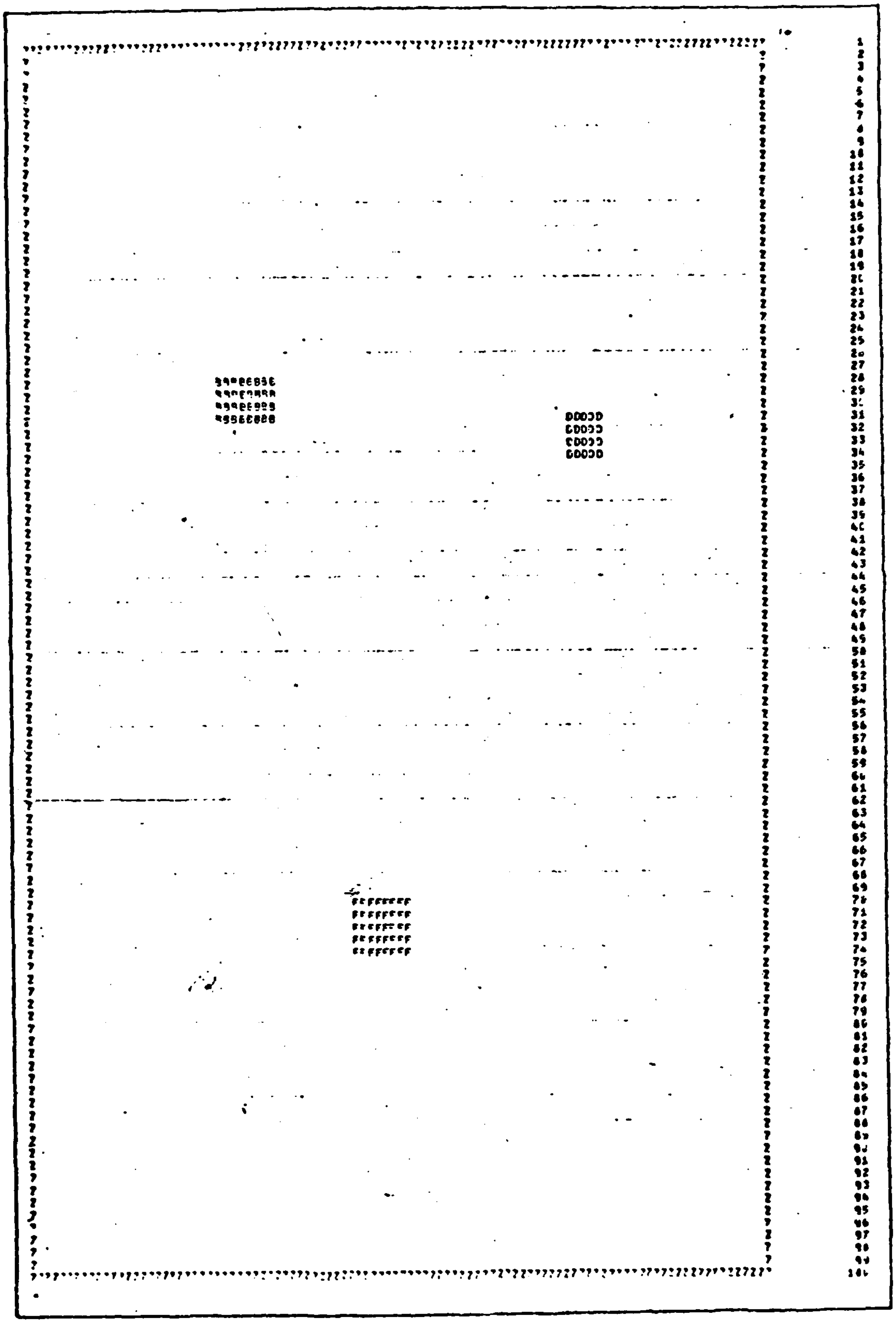


Figure 3-9. Results of Size Test

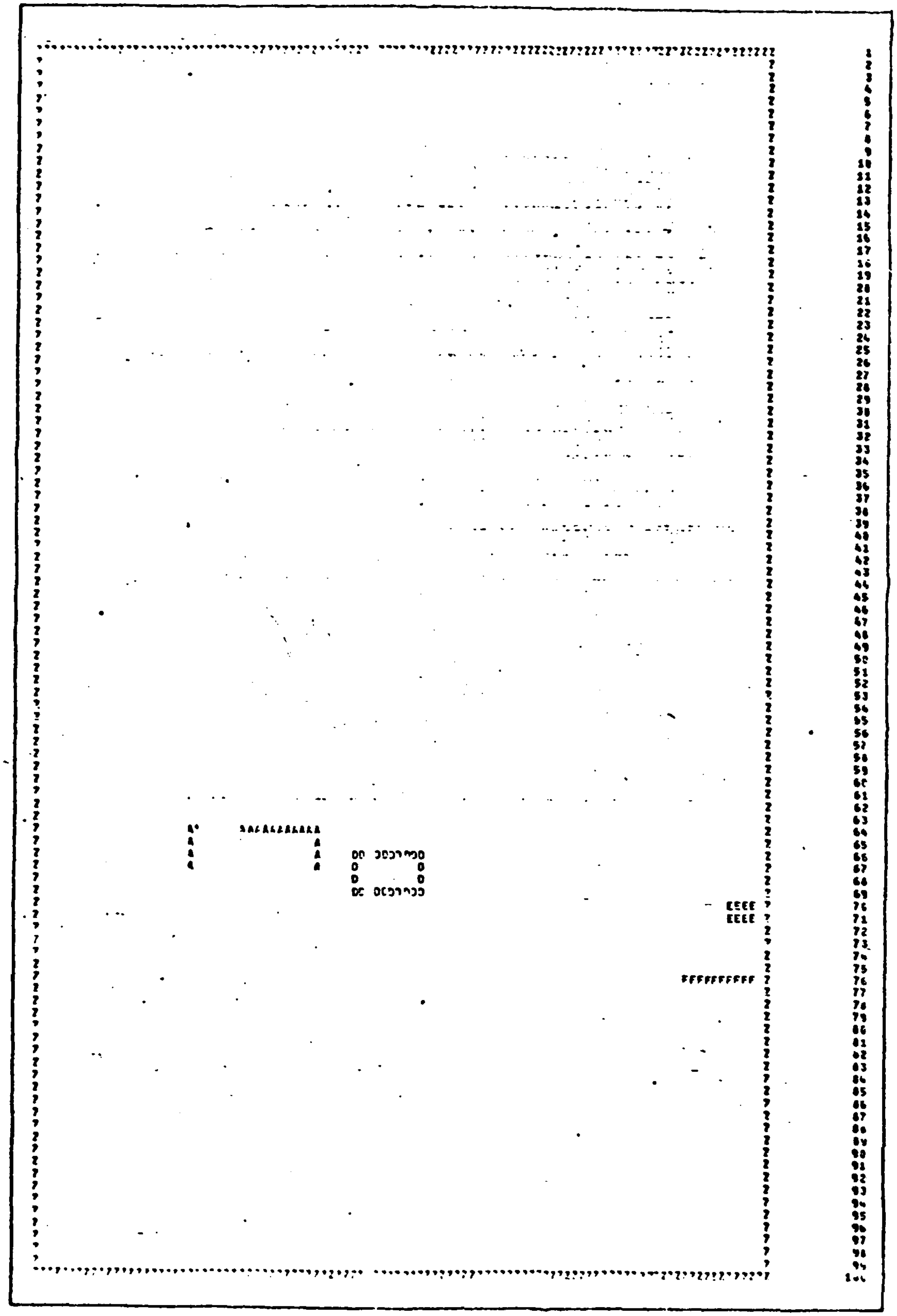


Figure 3-10. Special Case

lies in the analysis of the results achieved by using different percentages. Seventy-five percent was the final percentage which seemed to give acceptable results. What that means is that, if 75% of the total pixels which theoretically should lie on the circumference of an object of a certain size are detected during the whole process, that object is a target. As an example, if an object is five pixels by five pixels, the total number of pixels defining its circumference are 20. Out of 20 pixels, if (0.75 x 20 =) 15 pixels are detected, that object would be termed as a target.

So the objects which fulfill all the criteria explained in the last two chapters are termed as targets. Each one of them is given a separate letter identification and printed out by computer. Figure 3.11 is a final output of Figure 3.10. The reason why some of the letters are missing is that, after edge detection and closedness tests, each cluster of pixels is given a separate letter identifier. As each test is applied, some of those clusters get rejected. So the letters identifying those rejected clusters are the ones which get missed.

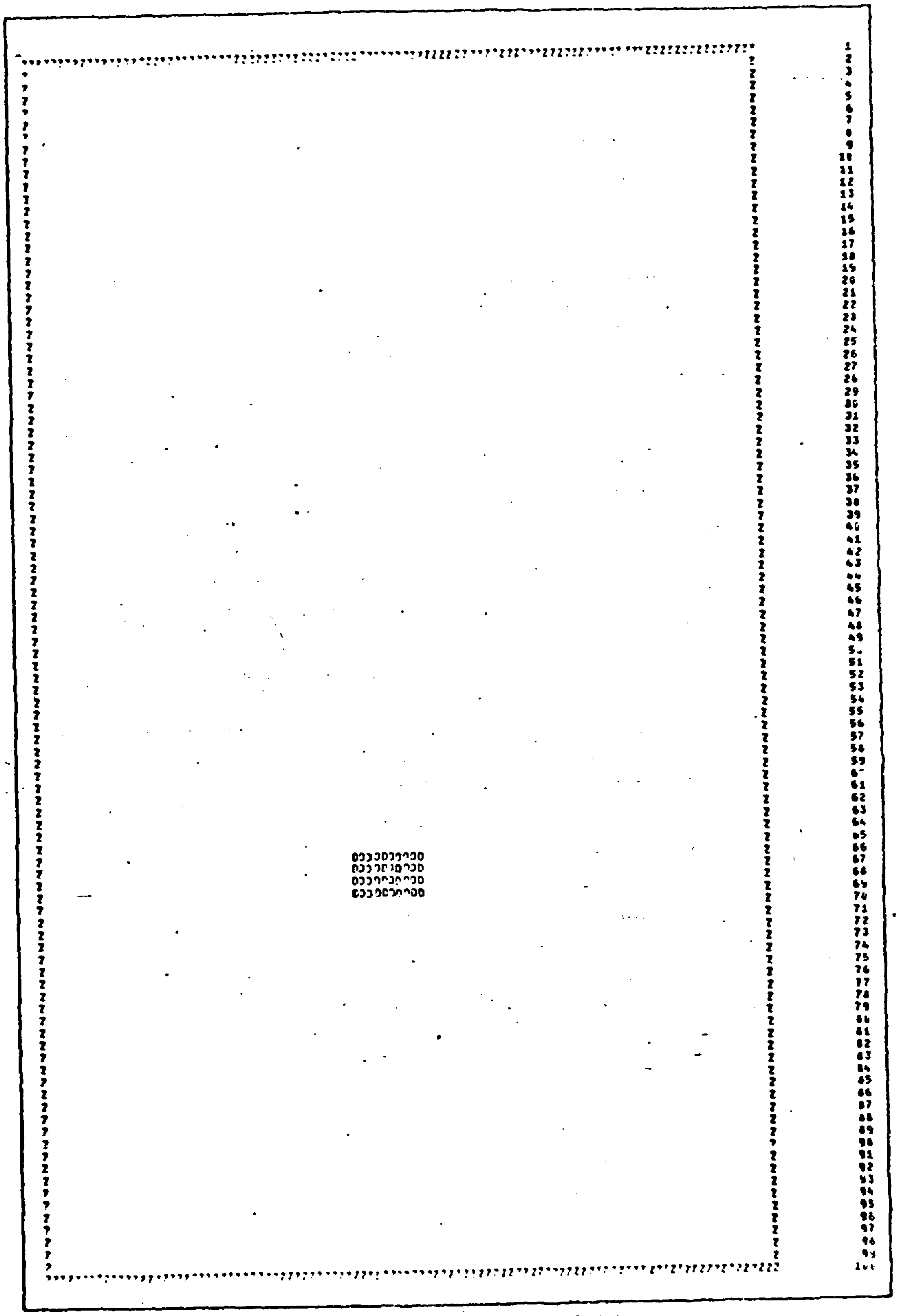


Figure 3-11. Final Processed Picture

IV. Results and Discussion

Introduction

The computer used for the purpose of this study was the CDC 6600 operated by the Aeronautical Systems Division at Wright-Patterson Air Force Base, and used by the Air Force Institute of Technology where this study was carried out. All of the programs were written in FORTRAN IV. A listing of the program and all the subroutines used can be found in Appendix D.

This chapter presents the results achieved and a critique of the algorithm in light of those results.

Results

A total of 325 pictures was processed for the purpose of this study. The data were provided by the Air Force Armament Laboratories, Eglin Air Force Base. The pictures contained various numbers of targets in different environmental conditions. The targets were tanks and trucks of different models, some unknown objects, and fire. For 199 pictures, the information of the location and kinds of targets in each individual picture was provided. Therefore, the accuracy scoring results presented here are only for 199 pictures.

The results achieved were analyzed on the basis of two parameters; namely, the probability of detection and the

probability of false alarm. Both of these parameters are a measure of ability and accuracy of the algorithm to detect targets in a scene. For the purpose of this thesis, the parameters have been defined as follows.

$$\text{Probability of Detection} = \frac{\text{Number of Targets Detected}}{\text{Total Number of Targets Present}}$$

$$\text{Probability of False Alarm} = \frac{\text{Number of False Targets Detected}}{\text{Total Number of Pictures}}$$

The detected targets are those targets which are detected by the process and are also known to be present. The false targets are those targets which are detected by the process, but are known to be not present. The probability of false alarm as defined above gives the number of false alarms detected per picture. It can also be defined as

$$\text{Probability of False Alarm} = \frac{\text{Number of False Targets Detected}}{\text{Total Number of Targets Detected}}$$

This probability of false alarm is that ratio of total detected targets which represents false detected targets.

were

For 199 pictures processed, the results obtained were:

Number of Targets Present	=	632
Number of Targets Detected	=	527
Number of False Targets Detected	=	57
Number of Pictures Processed	=	199

Therefore,

$$\text{Probability of Detection} = \frac{527}{632} = 83.39\%$$

Probability of False Alarm:

$$\text{by first definition} = \frac{57}{199} = 0.29 \text{ per picture}$$

$$\text{by second definition} = \frac{57}{57 + 527} = 9.76\%$$

The presence and absence of targets in the actual scenes were considered strictly on the basis of information provided. There were, however, instances where a target was known to be present, but the grey scale pictures showed absolutely no such indication. Similarly, there were other instances when a target was not shown, but the pictures showed the target without any doubt. In the latter case, the detected targets have been included in false alarms. But it may be the case that some of them are classified countermeasures used during data collection. The algorithm, as it is, is susceptible to countermeasures. But because of lack of information provided, no endeavors could be made to adapt the algorithm to reject such targets.

The algorithm uses 3 x 3 windows at two stages -- the image enhancement and edge detection. The use of windows has an inherent disadvantage. The disadvantage is that the rows lying on the upper and lower boundaries and the columns on the right

and left boundaries of the picture cannot be processed. Using the windows twice leaves out two rows or columns on each boundary without being processed. Therefore, the targets which are wholly constituted by pixels on these unprocessed rows or columns remain undetected. The detection of targets which are partly constituted by such pixels depends upon what ratio of total target is constituted of such pixels. This disadvantage need not have any effect as far as the processing in a real situation is concerned. In that case, frames are processed successively and a target appearing on boundary in one frame may well move into the frame with the movement of the sensor platform. But this disadvantage is nonetheless reflected in the results presented.

Discussion

In the real world, a sensor encounters infinite varieties of scenes. Useful mathematical modelling of these scenes, partially or completely, has not been accomplished as yet. Therefore, it is only logical not to use any fixed numbers for thresholding the scenes. It is rather essential that a dynamic thresholding be used which adapts itself to the different scenes as they are encountered. This was a major consideration in selecting the threshold in this algorithm. The threshold of mean plus variance is used in this process. Both mean and variance are those of the specific 100 x 100 pixel picture being processed. That makes the threshold dynamic. It does not use or assume any a priori knowledge

about the picture. Dynamic thresholding takes into consideration all kinds of situations like targets in both cold and hot backgrounds simultaneously.

A large amount of work already done in the field of target cueing is based upon one basic assumption; namely, some maximum possible target-size. Such an assumption, if logically made, does not seem to affect the detection much. But it is a serious limitation as far as range is concerned. Although it would work perfectly at long ranges, it will limit the detection seriously at closer ranges. The algorithm developed during this study does not make any such assumption. The algorithm does limit the ratio of two dimensions of the target, but there is no limit on the individual dimensions themselves. The man-made objects do have proportionate dimensions, and this is one of the basic features which differentiates them from many other objects in natural scenes.

The current algorithm does not go beyond detection. It does not classify the targets into different categories. Therefore, a logical extension of the algorithm will be to develop and include a classification routine. To cater for such potential future developments in the algorithm, it is essential that the basic features of the targets as present in the original picture be preserved. That is exactly what the algorithm does. The original intensity values of each individual pixel do remain unaltered after the algorithm.

Segmentation of targets after thresholding is one of the nice features of the algorithm. The segmentation should be accomplished in such a way that the targets can be separated no matter how close they are, although the segmentation becomes difficult as the targets get closer. The segmentation technique used by the algorithm separates two targets if they are one pixel away from each other. On the other hand, if two segments are connected by even one pixel, the technique does not separate them. The technique is simple and sensitive. It has worked very well on this data set. A detailed description of segmentation is given in Appendix C.

Another measure of quality and success of an algorithm is the size of the smallest target it can detect. It is important because it increases the range of effectiveness of the algorithm. The farther the targets are from the sensor, the smaller they will appear in the image. The smaller the targets an algorithm can detect, the greater the range of effectiveness of the algorithm. In other words, this is a measure of "sensitivity" of the algorithm analogous to the sensitivity of a receiver. The present algorithm can detect targets as small as 2 pixels x 2 pixels. This is the smallest possible size a target can have in the image. Smaller size than 2 x 2 would mean a point or a line, which is hard to perceive as a target. This small resolution would determine the effective range of the overall thermal imaging system of

which this algorithm will be a part. Because of the non-availability of optical parameters of the sensor such as field of view, range from the target and actual target size, it is not possible to say what range this 2 x 2 pixels resolution translates into. But it is certain that this is the best that can be achieved.

The detection of such small targets was made possible by the image-enhancement technique developed in this algorithm Image-enhancement not only improves the contrast of targets with respect to their background, it also enlarges the target size. This enlargement of size results in a disadvantage as far as separation of targets is concerned. The targets which lie very close to each other merge into each other to appear as one target after enlargement because of image-enhancement. In the final analysis, both the targets do get detected, but as one target. There were no such close targets in the analyzed data; therefore, no proper statistics such as how far the targets have to be from each other to be detected separately can be presented. The same effect is adversely felt when a target is very close to noise of large spatial size. Since the large unproportionate objects are not detected by the algorithm, image-enhancement merges the target into the noise and the target is missed.

The algorithm only detects the hot targets. Targets which are cooler than the background cannot be directly detected, although the same kind of thresholding and

processing as done for hot objects could be used for detecting the cold objects with a little modification.

The algorithm in its present form has no classification capability. The tanks and trucks cannot be separated from each other, and both of them cannot be separated from any other kind of target or noise that might appear as a target in a scene. Therefore, the algorithm is extremely susceptible to any kind of countermeasure that might be used against it.

Despite all these drawbacks and limitations, the algorithm seems to work extremely efficiently as far as final results are concerned.

V. Conclusions and Recommendations

Conclusions

The purpose of this study was to detect the targets (the man-made objects) from the low-resolution infrared imagery. The targets were defined to be those homogeneous proportionate areas in the picture which are constituted by pixels having intensity values above a certain threshold and the areas are bounded by the pixels having edge values greater than certain thresholds. Targets as small as 2 pixels x 2 pixels were considered and detected.

The whole process consisted of six steps:

(1) Image Enhancement

(2) Edge Detection

(3) Thresholding

(4) Connectedness Test

(5) Thinning

(6) Size and Proportion Test

A total of 325 pictures was analyzed during the development of this algorithm. The data set was prepared using different targets like tanks and trucks under different environmental and weather conditions. The results achieved were:

Probability of Detection = 83.39%

Probability of False Alarm = 9.76%

Because of time constraints, no work could be done on classification of targets. The algorithm detects potential targets, but is not capable of classifying the detected targets into tanks or trucks. Moreover, the time constraints did not allow the analysis of all pictures at different stages of detection. This kind of analysis will find at what stage the missed targets are being lost. This could lead to further improvements in the algorithm.

Recommendations

The algorithm, as it is, has given very encouraging results. Therefore, it seems reasonable to assume that the approach leads in the right direction. Because of time constraints, the study could not be carried on, but it is recommended that the work be continued in the following directions.

(1) Analysis of more data with the same algorithm to ascertain the performance of the algorithm under different environmental conditions. A rigorous analysis might result in a better understanding and thus pave the way for further development.

(2) Analysis of a large set of data under different stages of detection process. Such an analysis could pinpoint the stage which results in maximum and most-often misses. That stage can be further improved upon.

(3) Classification is the most obvious next step. A rigorous statistical analysis of the detected targets may result in determination of features capable of differentiating between a tank and a truck. A preliminary effort was made in this direction, but it did not prove successful. The pixels in a detected target were divided into four categories, depending upon their relative intensities. For this purpose, the mean and variance of every individual target was found. The division of the pixels was as follows:

A. pixels having intensities greater than mean plus variance

B. pixels having intensities between mean and variance

C. pixels having intensities between mean and mean minus variance

D. pixels having values less than mean minus variance.

Figure 5.1 shows a picture in which pixels in detected targets have been classified in this way. It is hoped that this kind of processing will make it possible to understand the textural qualities of the targets and thus classify them.

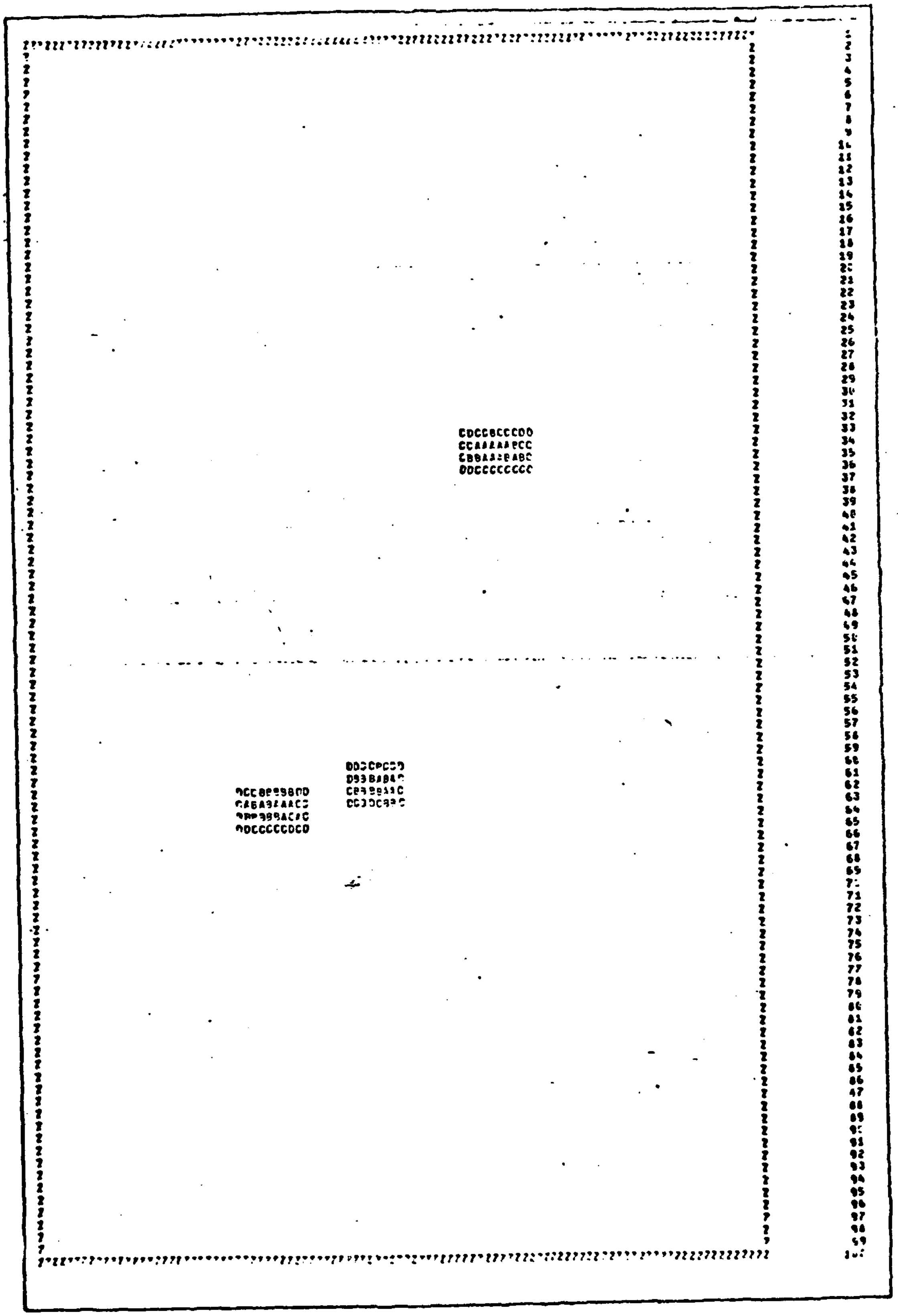


Figure 5-1. Pixel Classification

Bibliography

1. Gary M. Klein and Sahibsingh A. Dudan, "Locating Man-Made Objects in Low-Resolution Outdoor Scenes," Society of Photo and Instrument Engineers, Vol. 119, October 1977, pp. 278-283.

2. E.L. Hall, Computer Image Processing and Recognition, New York: Academic Press, 1979.

3. R. Duda and P. Hart, Pattern Classification and Scene Analysis, New York: John Wiley, 1973.

4. A. Rosenfeld, Picture Processing by Computer, New York: Academic Press, 1969.

5. R.A. Kirsch, "Computer Determination of the Constituent Structure of Biological Images," Computers and Biomedical Research, 4(3), June 1971, pp. 315-328.

6. W.K. Pratt, Digital Image Processing, New York: Wiley Interscience, 1978.

7. E. Argyle, "Techniques for Edge Detection," Proceedings of the IEEE, 59(2), February 1971, pp. 285-287.

8. L.S. Davis, "A Survey of Edge Detection Techniques," Computer Graphics and Image Processing, 4(3), September 1975, pp. 248-270.

9. Durga P. Panda, "Statistical Analysis of Some Edge Operators," TR 541, July 1977, Computer Science Technical Report Series, University of Maryland.

10. Ikram Escander Abdon, "Quantitative Methods of Edge Detection," Image Processing Institute, University of Southern California Report #830, July 1978. AD A059 124.

11. L.G. Roberts, "Machine Perception of Three-Dimensional Solids," Computer Methods in Image Analysis, Edited by J.K. Aggarwal, Richard O. Duda and Azriel Rosenfeld, IEEE Press, New York, 1977.

12. Dr. Matthew Kabrisky, personal conversation, Air Force Institute of Technology, Wright-Patterson AFB OH, June-September 1981.

APPENDIX A

Data

The data for 325 pictures were provided by Eglin Air Force Base, stored on magnetic tapes. The data were stored in the Standard Target and Background Information Library System (TABILS). A TABILS picture is made up of 102 four hundred character BCD records on the tape. The first two records contain values of different environmental parameters at the time the picture was taken. Tables A-1 and A-2 give the details of these parameters, the order and format in which they are stored. Records 3 through 102 contain the TABILS thermovision picture in counts. Values range between 0 and 1023. A value of 1023 counts represents "saturated data." A value greater than 1023 represents "no data" for that particular cell.

A picture is made up of 100 lines, each containing 100 words or 10,000 values. Each record on tape is 400 characters long, representing 100 words on the line. Each of these data records is stored as 100 I4. Alignment of the picture is as follows:

Line 1 = Back of 3-D picture as perceived by viewer

Line 100 = Front of 3-D picture as perceived by viewer

Word 1 = Left side of 3-D picture as perceived by viewer

Word 100 = Right side of 3-D picture as perceived by viewer.

To change the count values to radiance (watts/cm^2-Sr) the conversion factor slope and conversion factor intercept must be used from record 2.

$$y = mx + b$$

where

m = conversion factor slope

x = count value of picture cell

b = conversion factor intercept

y = radiance value of cell.

In the particular case of this thesis, the first two records of every picture were blank. Therefore, the conversion factors of both slope and intercept were not available. This fact led to the processing of pictures not with the radiance values for each cell, but the count values. Since the transformation from radiance to count values is linear, therefore, no contrast reduction was assumed. Rather, it was advantageous to manipulate small numbers on the computer.

TABLE A-1

Labelling

Parameter Name	Format	Comments
Target Code	I3	Coded Item
Background	2A10	
Countermeasures Code	I3	Coded Item
Band Code	I3	Coded Item
Collection Agency	2A10	
Test Side Location Code	I3	Coded Item
Mission Date	A6	YYMMDD
Weather Code	A1	G (Good) or B (Bad)
Weather Description	4A10	
Measuring Instrument	2A10	
Target Viewing Angle*	I6	in Degrees
Depression Angle**	I6	in Degrees
Range	I6	in Meters
Scene Description	7A10	
RPM (Target)	I6	
Prior Vehicle State (Target)	A10	Cold, exercised or idled
Prior History	4A10	
Relative Bearing Angle***	I6	in Degrees
Wind Direction	I6	in Degrees
Visible Transmission	I6	in Percent
IR Transmission	I6	in Percent
IR Transmission Band Code	I6	Coded Item
Pyranometer Reading	I6	in Watts/meter^2
Pyrheliometer Reading	I6	in Watts/meter^2
Filler	I6	95 Blank Characters

* Angle measured from the target pointing vector to the ground projection of a line from the target to measuring instrument (positive clockwise)

** Angle measured from the horizontal to the measuring instrument pointing vector (positive down)

*** Angle measured from true north to the ground projection of a line from the measuring instrument to the target (positive clockwise)

TABLE A-2

Record2 Labelling

Parameter Name	Format	Comments
Picture Time	F10.3	HHMMSS.SSS
Conversion Factor - Slope	F20.15	
Conversion Factor - Intercept	F20.15	
Picture Comments	7A10	
Horizontal Resolution	F6.3	in Milliradians
Vertical Resolution	F6.3	in Milliradians
Relative Humidity	F10.2	in Percent
Wind Speed	F10.2	in Knots
Air Temperature	F10.1	in Degrees Centigrade
Barometric Pressure	F10.0	in Millibars
Rain Rate	F10.2	in Millimeters/hour
Snow Temperature	F10.2	in Degrees Centigrade
Soil Moisture Content	F10.2	in Percent
Visibility	F10.2	in Kilometers
Filler		188 Blank Characters

APPENDIX B

Gray Scale

The pictures from the tape were printed on a line printer using a combination of overprinted characters. Because of the size of the picture (100 x 100), the line printer posed no restrictions in printing the complete horizontal line completely. For larger sizes of pictures, the maximum characters a line printer can print in a horizontal line may be a serious limitation.

A maximum of eight overprinted characters for maximum intensity was used. The density codes (i.e., the overprinted character combinations) are given in Table B-1.

As shown in Table B-1, the combinations of overprinted characters appear for values between 0 and 1. Therefore, the whole picture values to be printed were linearly scaled down between 0 and 1.

$$y = mx + b$$

x = count value of picture cell

m = 1.0 / (max - min)

b = m * min

y = scale down value of picture cell

max = maximum count value of picture

min = minimum count value of picture.

This method of printing the pictures produced a variant of the nominally expected picture. The hotter points in the picture have high intensity and thus would be expected to appear lighter in the picture. But the combinations of overprinted characters is such that the higher density, i.e., the hotter points, appear as dark points in the printed picture. This representation of the data was found to be perfectly satisfactory.

TABLE B-1

Overprinted Character Combinations	Intensity Value
Blank	0.00
-	0.15
=	0.22
+	0.25
>	0.29
I	0.33
z	0.37
x	0.40
A	0.42
M	0.45
0-	0.53
0=	0.56
0+	0.60
0+´	0.64
0+´•	0.67
0+´•=	0.79
0+´•-	0.85
0x´•HC	0.89
0x´•HB	0.93
0x´•HBV	0.97
0x´•HBVA	1.00

APPENDIX C

Segmentation

After the connectedness test has been accomplished, the pictures contain only a few scattered blobs that have to be tested to determine if they are true representatives of actual targets. Before this task can be accomplished, it is essential that these blobs be numbered separately so that they can be individually analyzed. This process of separating the blobs for computer use is called segmentation.

For the purpose of this study, the segmentation was achieved in two steps. The basic purpose of separating the blobs was served by the first step, but in certain cases all the pixels in a blob are not numbered similarly by the first step. Therefore, the second step is used in order to keep the numbers the same.

After the connectedness test, most of the pixels in the matrix are left with a value of zero. Only those pixels which form a part of the blobs have a non-zero value. So, essentially, what segmentation has to achieve is to give a particular integer value to all non-zero pixels which are connected together. Another task which segmentation must perform simultaneously is to increment or change that particular integer value once it detects a blob which is not connected to the previous one.

Figure C-1 is a picture after connectedness tests have been performed. In the first step, the picture is scanned from left to right, starting from the third row. The first two rows are left because they have not been processed because of use of 3 x 3 windows. Before the start of the scan, a counter is set to zero and another matrix (integer matrix) with the same dimensions as the picture matrix is initialized with all pixels having a zero value. During the scan of the picture matrix, whenever a non-zero pixel is detected, the scanning stops. The upper and left pixels in the integer matrix, corresponding to upper and left pixels of the detected non-zero pixel in the picture matrix, are checked for non-zero values. If any one of them is non-zero, the same value is given to the pixel in the integer matrix corresponding to a detected non-zero pixel in the picture matrix. If both of them are zero, the counter is incremented by 1 and the value of the counter is assigned to the pixel in the integer matrix corresponding to a detected non-zero pixel in the picture matrix. The scanning resumes and the process continues until the picture is finished. Figure C-2 is the representation of Figure C-1 after the first step of segmentation.

It should be noticed that not all the points in all the blobs are numbered the same. This is because the first step does not consider all possible shapes a blob can assume. Therefore, a second scan of the integer matrix is performed.

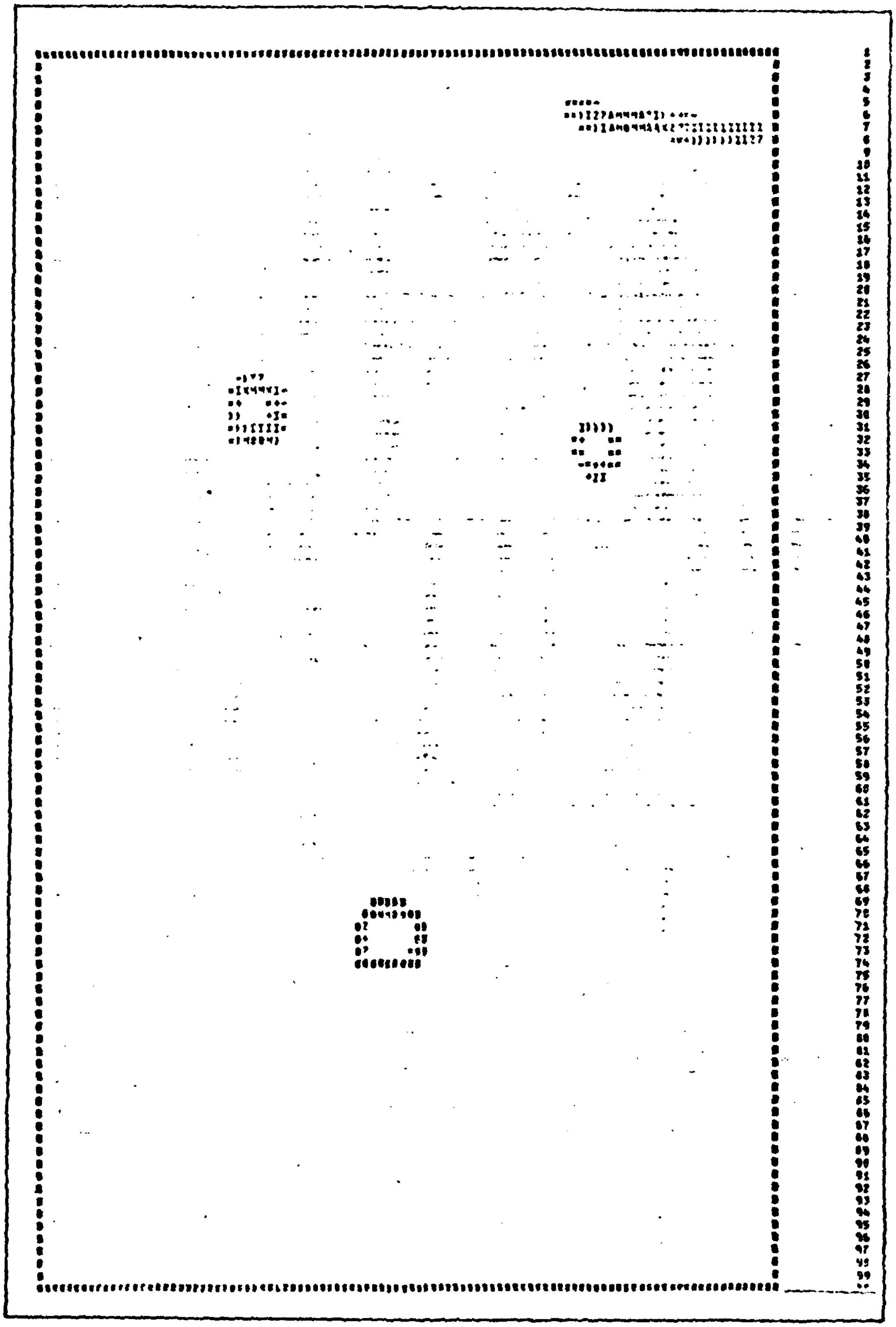


Figure C-1. Picture After Closedness Test

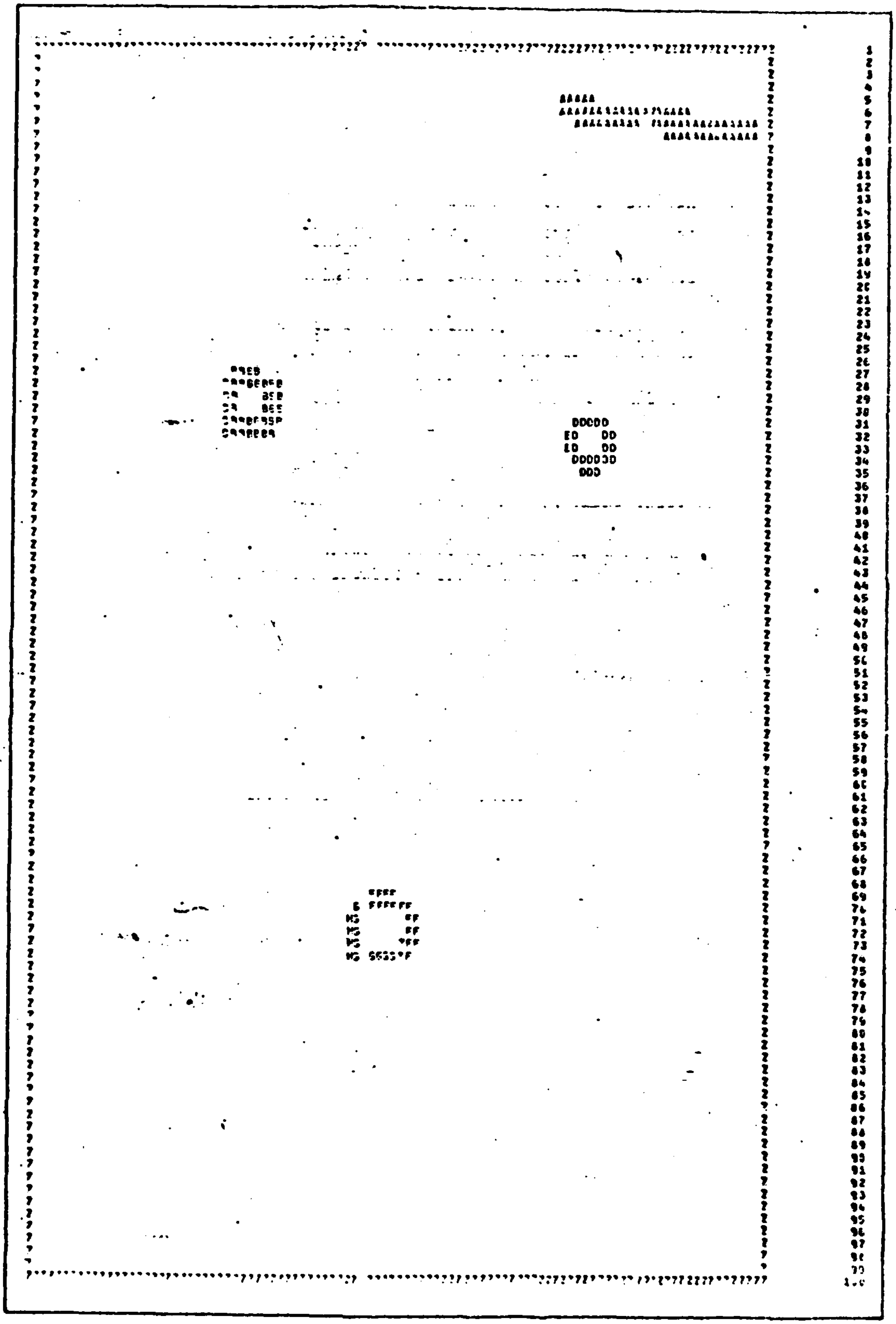


Figure C-2. Picture After 1st Step of Segmentation

During the scan of the integer matrix, when a non-zero pixel value is encountered, the scan stops. The processing is done in the following stages:

(a) (i) If the pixel on the right is zero or has the same value as the one under consideration, to to the next stage of processing.

(ii) If the pixel on the right is not zerc and does not have the same value as the pixel under consideration, assign the minimum of the two values to both of them and to to the next stage.

(b) Perform the same steps as in (a), but this time with the pixel on the left and the detected non-zero pixel.

(c) Perform the same steps as in (a), but with the lower pixel and the detected non-zero pixel.

Before the scan starts, a counter is set to zero. Whenever a situation is encountered when a non-zero pixel does not have the same value as one of its left, right, or lower non-zero neighbors, the counter is incremented by one. After the (a), (b), and (c) stages have been completed, the scan resumes until the total integer matrix has been scanned. After the total matrix is scanned, the value of the counter is checked. If the value is non-zero, the whole scanning of the integer matrix is repeated with the counter reset to zero. This process is repeated until a zero value in the

counter is encountered after a complete scan of the matrix. Figure C-3 is the final segmented picture.

Figures C-4 and C-5 are the flow diagrams for the two segmentation steps, respectively.

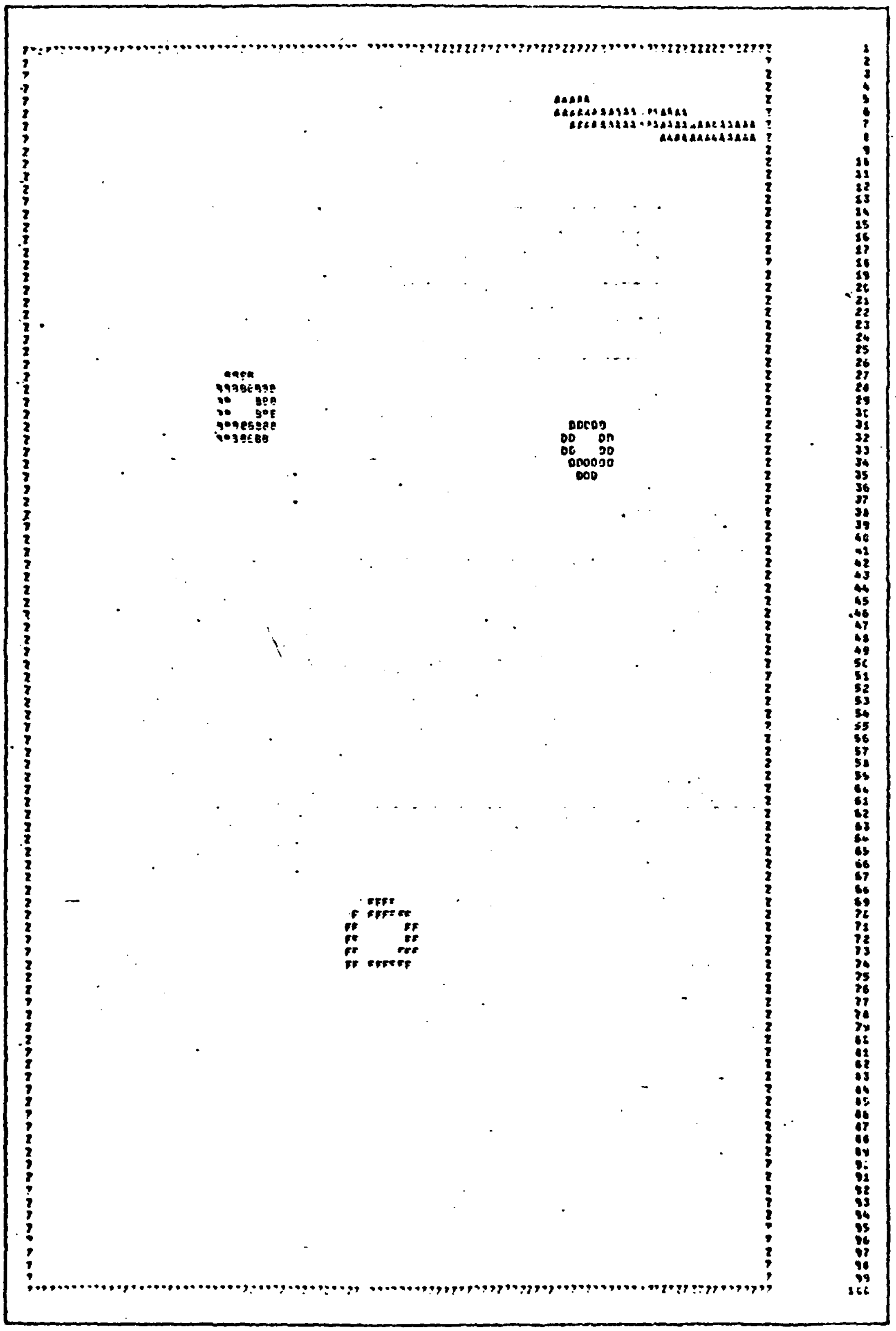


Figure C-3. Picture After Segmentation

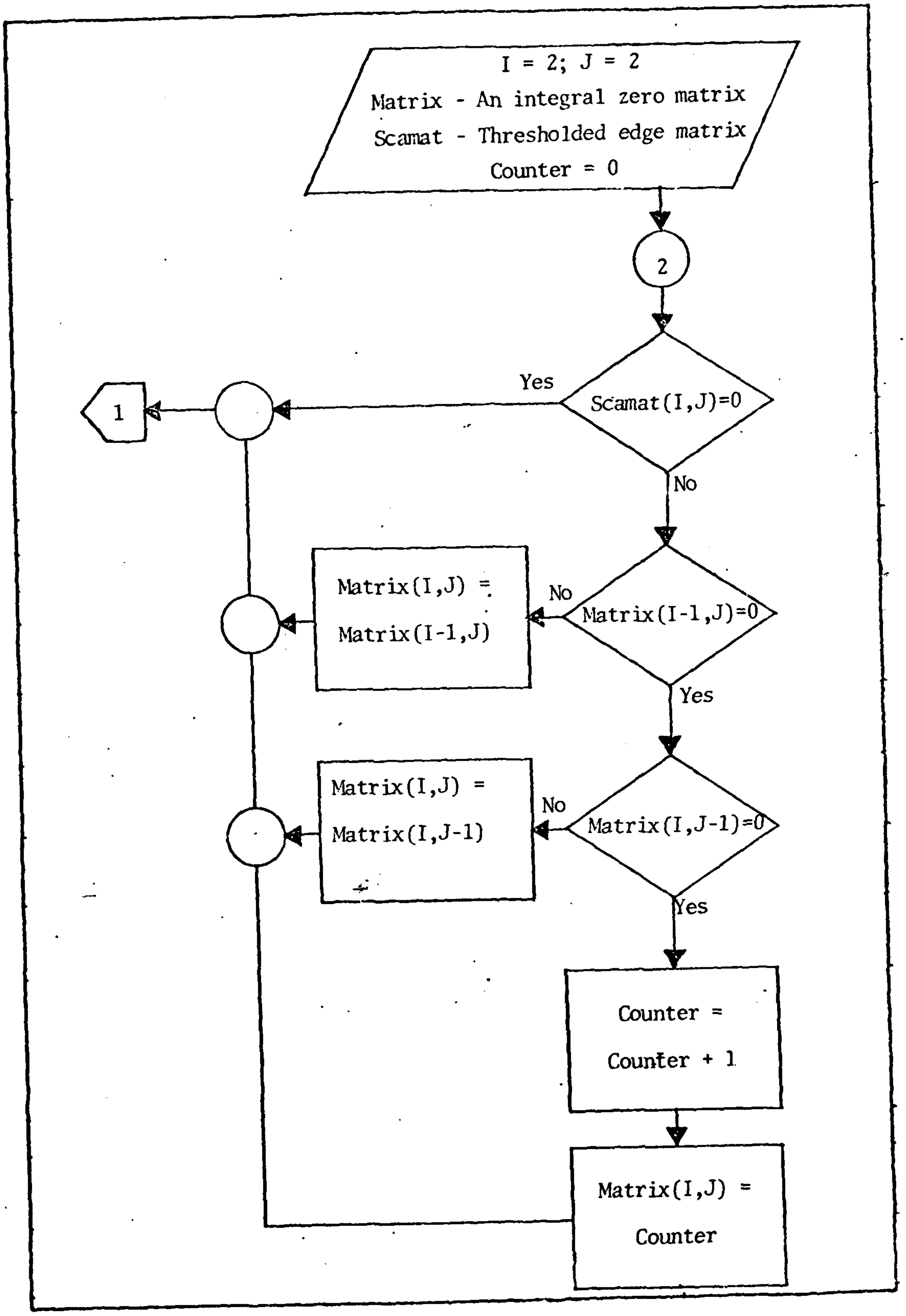
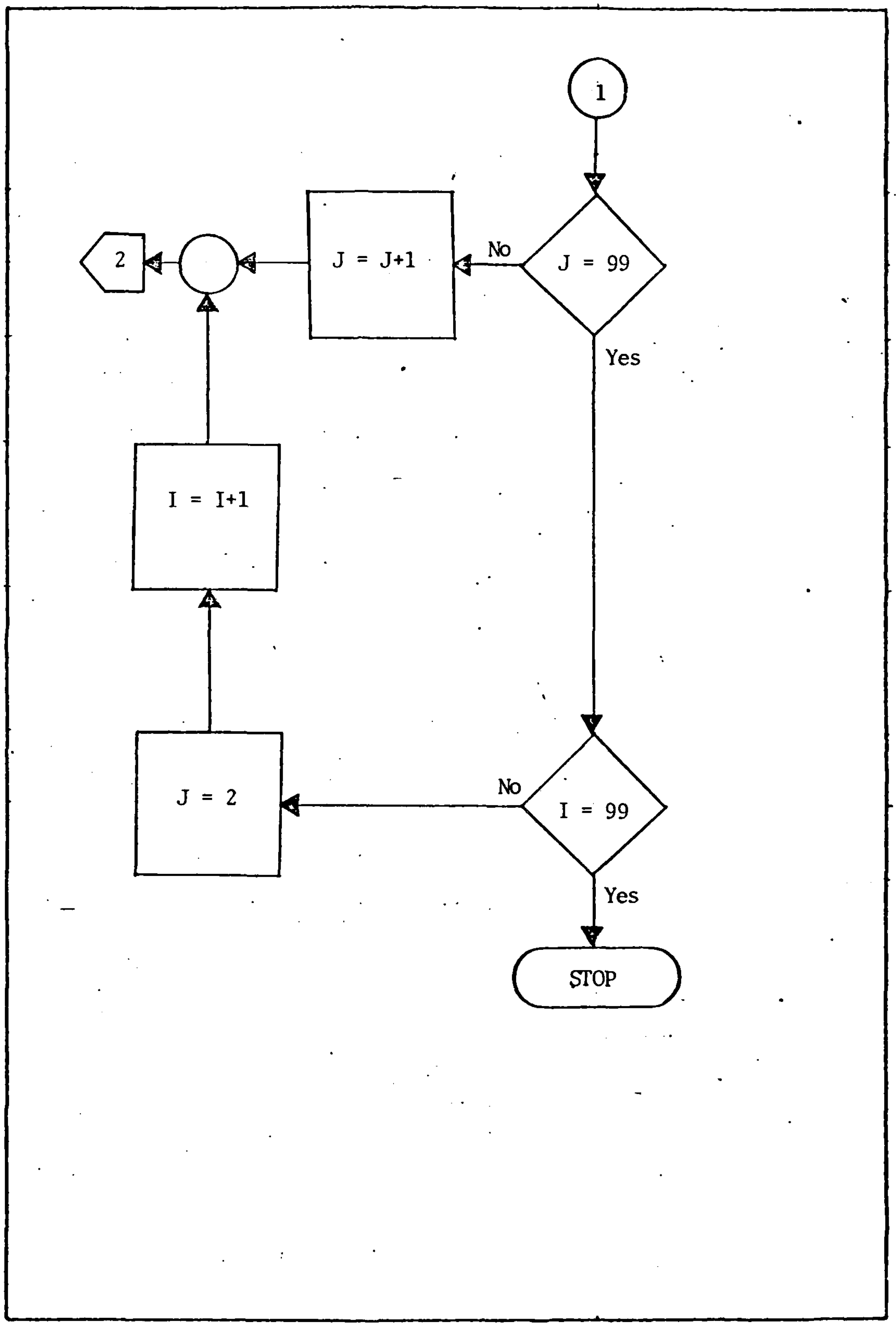


Figure C-4. 1st Step of Segmentation


1
J = 99
No
Yes
J = J+1
2
I = I+1
J = 2
No
I = 99
Yes
STOP

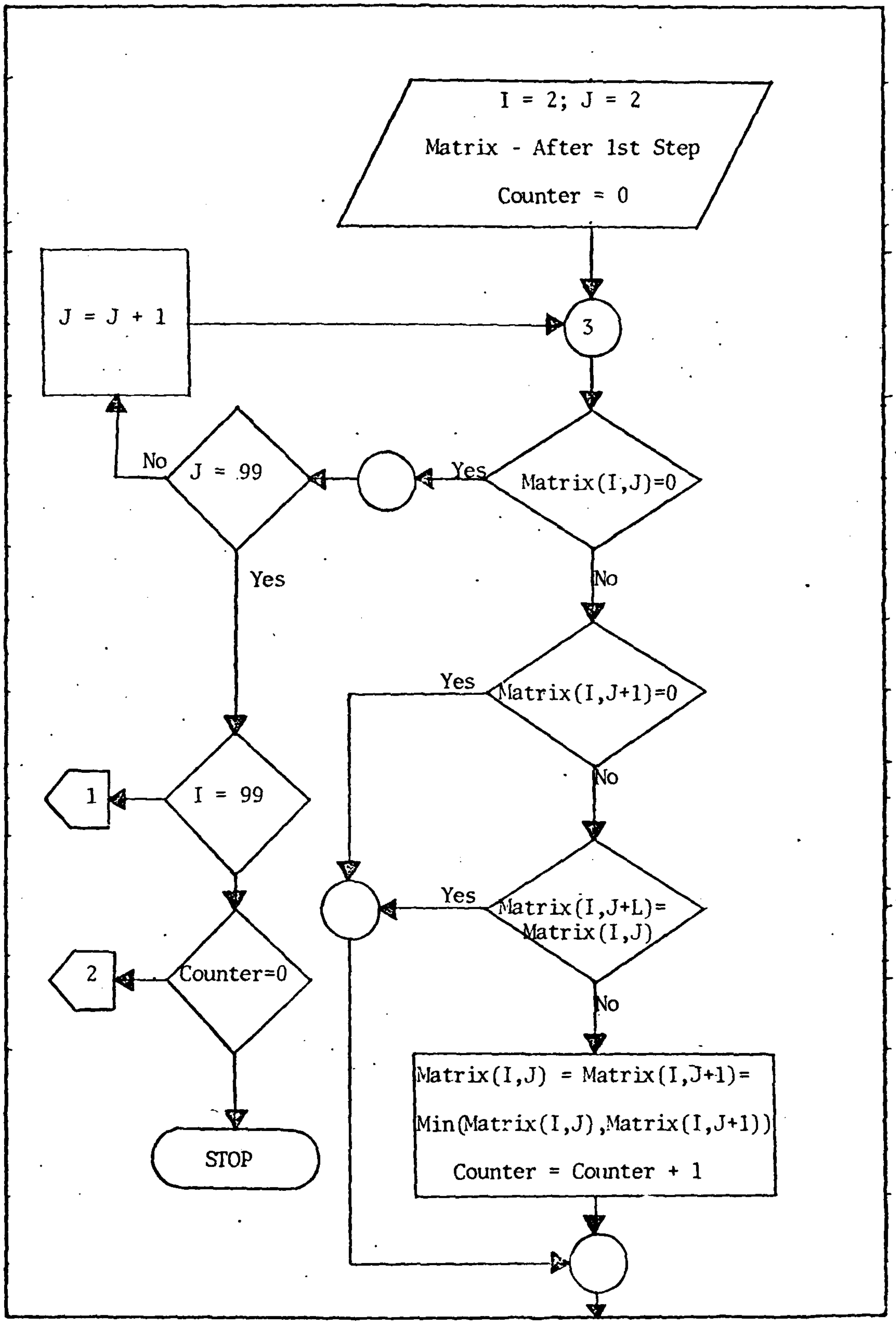
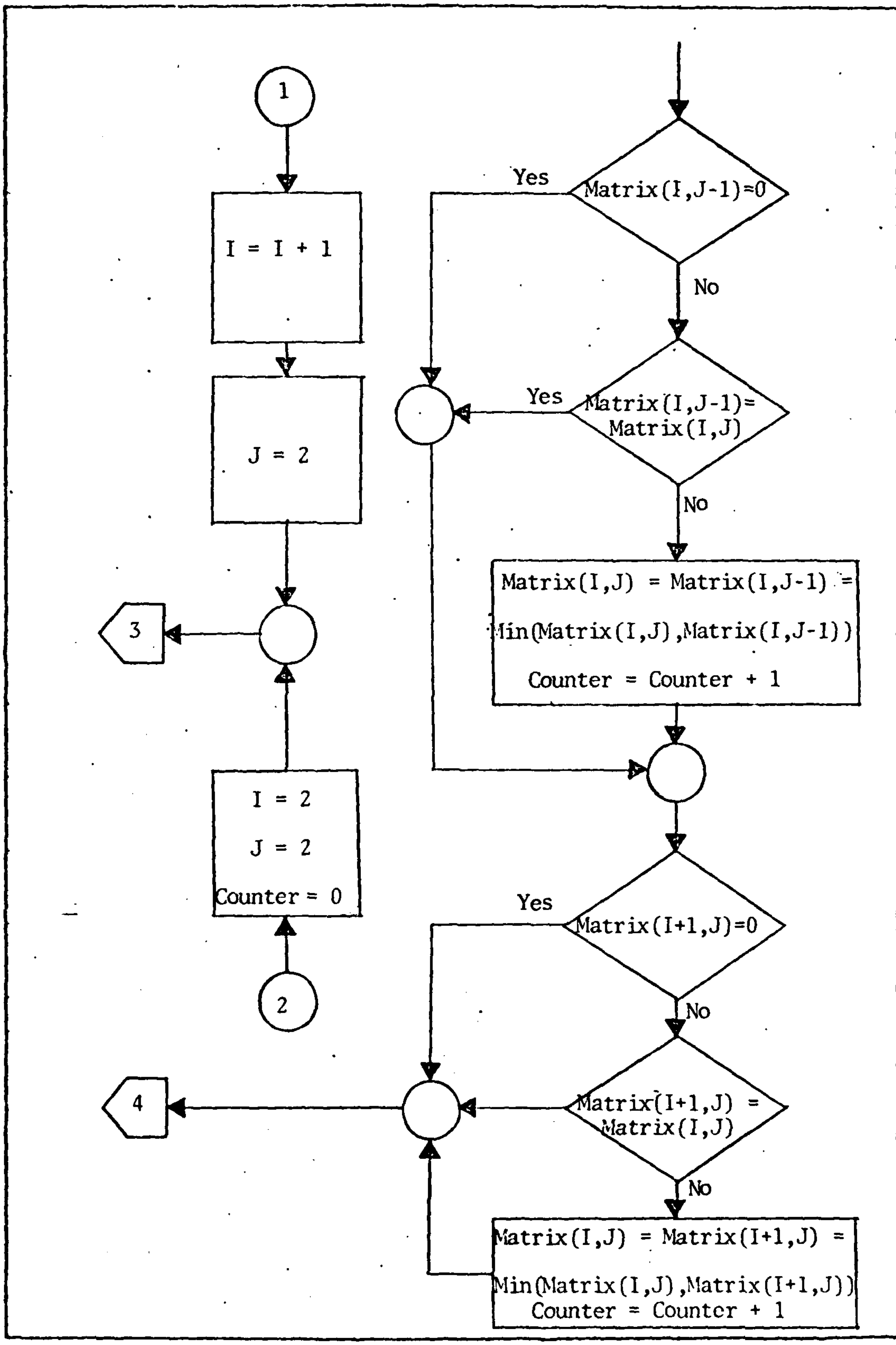


Figure C-5. 2nd Step of Segmentation


1
I = I + 1
J = 2
3
I = 2
J = 2
Counter = 0
2
4
Matrix(I,J-1)=0
Yes
No
Matrix(I,J-1)=
Matrix(I,J)
Yes
No
Matrix(I,J) = Matrix(I,J-1) =
Min(Matrix(I,J),Matrix(I,J-1))
Counter = Counter + 1
Matrix(I+1,J)=0
Yes
No
Matrix(I+1,J) =
Matrix(I,J)
No
Matrix(I,J) = Matrix(I+1,J) =
Min(Matrix(I,J),Matrix(I+1,J))
Counter = Counter + 1

APPENDIX D

Program Listing

The following pages present the source listing of the final program used to detect targets.

```
*******************************************************************
*FOLLOWING IS THE PROGRAM WRITTEN FOR DETECTING TARGETS FROM LOW*
*RESOLUTION IR IMAGERY. THE PICTURES ARE IN THE FORM OF A 100*   *
*100 MATRICES. THE DATA WAS PROVIDED BY EGLIN AIR FORCE BASE.    *
*THE DATA TAPE IS WITH THE ASD COMPUTER CENTER UNDER PROBLEM #   *
*T75'6[illegible] AND VSN=X[illegible]. IT IS ATTACHED AS TAPE5 BEFORE THE       *
*PROGRAM IS RUN. THIS PROGRAM THEN EXECUTES THE ALGORITHM DEVEL-*
*OPED BY FLT. LT. HAMADANI AND PRINTS THE RESULTS OF EACH STAGE *
*IN THE ALGORITHM IN GRAY-SCALE AS CONTAINED IN SUBROUTINE      *
*CCGRAY.DARK POINTS ARE THE HOT POINTS.                          *
*       -----------------------------                            *
*******************************************************************

*****************************************************
*                LIST OF VARIABLES                  *
* MATRIX= ORIGINAL PICTURE MATRIX                   *
* JATRIX= IMAGE ENHANCED PICTURE MATRIX             *
* NATRIX= EDGE MATRIX OF JATRIX                     *
* SCAMAT= ANY MATRIX,TO BE PRINTED,SCALED BETWEEN   *
*         0 AND 1                                   *
* MINI= UPPERMOST ROW OF A CONNECTED CLUSTER        *
* MINJ= LEFTMOST COLUMN OF A CONNECTED CLUSTER      *
* MAXI= LOWERMOST ROW OF A CONNECTED CLUSTER        *
* MAXJ= RIGHTMOST COLUMN OF A CONNECTED CLUSTER     *
* ITEST= A DUMMY VARIABLE WHICH DETERMINES WHETHER*
*      OR NOT A CLUSTER REPRESENTS A TARGET         *
* NP= # OF NON-ZERO PIXELS IN A TARGET BOUNDARY     *
*****************************************************
PROGRAM ZAINAB(INPUT,OUTPUT,TAPE5=/400)
COMMON SCAMAT(100,100)
DIMENSION MATRIX(100,100),NATRIX(100,100),SCAMAT(100,100),
+    JATRIX(100,100),PATRIX(100,100),
+   MINI(30),MINJ(30),MAXI(30),MAXJ(30),ITEST(30),NP(30)
*******************************************************
* THE PICTURE IS READ FROM TAPE5. FIRST TWO RECORDS  *
* ARE BLANK, THEREFORE THEY ARE READ INTO DUMMY VAR- *
* IABLES. NEXT 100 RECORDS,WITH 100 VALUES EACH, ARE *
* READ INTO THE MATRIX.                              *
*******************************************************
READ(5,1)A
READ(5,1)B
1    FORMAT(I4)
     DO 10 I=1,100
     READ(5,905)(MATRIX(I,J),J=1,100)
10   CONTINUE
905  FORMAT(100I4)
*************************************************************
* IMAGE ENHANCED MATRIX(JATRIX) IS FORMED FROM MATRIX.     *
* EXTREME ROWS AND COLUMNS CANNOT BE EVALUATED BECAUSE OF*
* USE OF 3*3 WINDOWS.                                       *
*************************************************************
```

```
      DO 55 I=2,99
      DO 50 J=2,99
      IA=IABS((1[illegible]*(MATRIX(I-1,J-1)+MATRIX(I-1,J)+MATRIX(I-1,J+1)))-(
     +   MATRIX(I,J+1)+MATRIX(I+1,J+1)+MATRIX(I+1,J)+MATRIX(I+1,J-1)+
     +   MATRIX(I,J-1)))
      IB=IABS((1[illegible]*(MATRIX(I-1,J)+MATRIX(I-1,J+1)+MATRIX(I,J+1)))-(
     +   MATRIX(I+1,J+1)+MATRIX(I+1,J)+MATRIX(I+1,J-1)+MATRIX(I,J-1)+
     +   MATRIX(I-1,J-1)))
      IC=IABS((1[illegible]*(MATRIX(I-1,J+1)+MATRIX(I,J+1)+MATRIX(I+1,J+1)))-(
     +   MATRIX(I+1,J)+MATRIX(I+1,J-1)+MATRIX(I,J-1)+MATRIX(I-1,J-1)+
     +   MATRIX(I-1,J)))
      ID=IABS((1[illegible]*(MATRIX(I,J+1)+MATRIX(I+1,J+1)+MATRIX(I+1,J)))-(
     +   MATRIX(I+1,J-1)+MATRIX(I,J-1)+MATRIX(I-1,J-1)+MATRIX(I-1,J)+
     +   MATRIX(I-1,J+1)))
      IE=IABS((1[illegible]*(MATRIX(I+1,J+1)+MATRIX(I+1,J)+MATRIX(I+1,J-1)))-(
     +   MATRIX(I,J-1)+MATRIX(I-1,J-1)+MATRIX(I-1,J)+MATRIX(I-1,J+1)+
     +   MATRIX(I,J+1)))
      IF=IABS((1[illegible]*(MATRIX(I+1,J)+MATRIX(I+1,J-1)+MATRIX(I,J-1)))-(
     +   MATRIX(I-1,J-1)+MATRIX(I-1,J)+MATRIX(I-1,J+1)+MATRIX(I,J+1)+
     +   MATRIX(I+1,J+1)))
      IG=IABS((1[illegible]*(MATRIX(I+1,J-1)+MATRIX(I,J-1)+MATRIX(I-1,J-1)))-(
     +   MATRIX(I-1,J)+MATRIX(I-1,J+1)+MATRIX(I,J+1)+MATRIX(I+1,J+1)+
     +   MATRIX(I+1,J)))
      IH=IABS((1[illegible]*(MATRIX(I,J-1)+MATRIX(I-1,J-1)+MATRIX(I-1,J)))-(
     +   MATRIX(I-1,J+1)+MATRIX(I,J+1)+MATRIX(I+1,J+1)+MATRIX(I+1,J)+
     +   MATRIX(I+1,J-1)))
      NAX=0
      JATRIX(I,J)=MAX0(IA,IB,IC,ID,IE,IF,IG,IH,1)
50    CONTINUE
55    CONTINUE
      *************************************************************
      * MEAN AND VARIANCE OF THE IMAGE ENHANCED PICTURE IS        *
      * EVALUATED.                                                *
      *************************************************************
      CALL THRESH(JATRIX,MEAN1,VAR1)
      *********************************************************
      * THE EDGE MATRIX FOR THE IMAGE ENHANCED PICTURE IS *
      * EVALUATED. AGAIN THE EXTREME ROWS AND COLUMNS OF  *
      * THE IMAGE ENHANCED PICTURE HAVE BEEN LEFT OUT.    *
      *********************************************************
      DO 45 I=3,98
      DO 45 J=3,98
      IA=IABS((5*(JATRIX(I-1,J-1)+JATRIX(I-1,J)+JATRIX(I-1,J+1)))-(3*(
     +   JATRIX(I,J+1)+JATRIX(I+1,J+1)+JATRIX(I+1,J)+JATRIX(I+1,J-1)+
     +   JATRIX(I,J-1))))
      IB=IABS((5*(JATRIX(I-1,J)+JATRIX(I-1,J+1)+JATRIX(I,J+1)))-(3*(
     +   JATRIX(I+1,J+1)+JATRIX(I+1,J)+JATRIX(I+1,J-1)+JATRIX(I,J-1)+
     +   JATRIX(I-1,J-1))))
      IC=IABS((5*(JATRIX(I-1,J+1)+JATRIX(I,J+1)+JATRIX(I+1,J+1)))-(3*(
     +   JATRIX(I+1,J)+JATRIX(I+1,J-1)+JATRIX(I,J-1)+JATRIX(I-1,J-1)+
     +   JATRIX(I-1,J))))
      ID=IABS((5*(JATRIX(I,J+1)+JATRIX(I+1,J+1)+JATRIX(I+1,J)))-(3*(
     +   JATRIX(I+1,J-1)+JATRIX(I,J-1)+JATRIX(I-1,J-1)+JATRIX(I-1,J)+
     +   JATRIX(I-1,J+1))))
```

```
      IE=IABS((5*(JATRIX(I+1,J+1)+JATRIX(I+1,J)+JATRIX(I+1,J-1)))-(3*(
     +   JATRIX(I,J-1)+JATRIX(I-1,J-1)+JATRIX(I-1,J)+JATRIX(I-1,J+1)+
     +   JATRIX(I,J+1))))
      IF=IABS((5*(JATRIX(I+1,J)+JATRIX(I+1,J-1)+JATRIX(I,J-1)))-(3*(
     +   JATRIX(I-1,J-1)+JATRIX(I-1,J)+JATRIX(I-1,J+1)+JATRIX(I,J+1)+
     +   JATRIX(I+1,J+1))))
      IG=IABS((5*(JATRIX(I+1,J-1)+JATRIX(I,J-1)+JATRIX(I-1,J-1)))-(3*(
     +   JATRIX(I-1,J)+JATRIX(I-1,J+1)+JATRIX(I,J+1)+JATRIX(I+1,J+1)+
     +   JATRIX(I+1,J))))
      IH=IABS((5*(JATRIX(I,J-1)+JATRIX(I-1,J-1)+JATRIX(I-1,J)))-(3*(
     +   JATRIX(I-1,J+1)+JATRIX(I,J+1)+JATRIX(I+1,J+1)+JATRIX(I+1,J)+
     +   JATRIX(I+1,J-1))))
      NAX=0
      NAX=MAX0(IA,IB,IC,ID,IE,IF,IG,IH)
      NATRIX(I,J)=MAX0(1,NAX)
 40   CONTINUE
 45   CONTINUE
C***************************************************************
C* FIRST THE MEAN AND VARIANCE OF THE EDGE MATRIX IS EVALUA-*
C* TED. SECONDLY THE MINIMA AND MAXIMA FOR ALL THE MATRICES *
C* AVAILABLE ARE CALCULATED.                                 *
C* MAX= MAXIMUM VALUE IN IMAGE ENHANCED PICTURE              *
C* MIN= MINIMUM VALUE IN IMAGE ENHANCED PICTURE              *
C* MAX1= MAX. VALUE IN ORIGINAL PICTURE                      *
C* MIN1= MIN. VALUE IN ORIGINAL PICTURE                      *
C* MAX2= MAX. VALUE IN THE EDGE MATRIX                       *
C* MIN2= MIN. VALUE IN THE EDGE MATRIX                       *
C***************************************************************
      CALL THRESH(NATRIX,DEAN2,VAR2)
      MAX=JATRIX(2,2)
      MAX1=MATRIX(1,1)
      MAX2=NATRIX(3,3)
      MIN=JATRIX(2,2)
      MIN1=MATRIX(1,1)
      MIN2=NATRIX(3,3)
      DO 25 I=3,98
      DO 20 J=3,98
      MIN=MIN0(MIN,JATRIX(I,J))
      MIN1=MIN0(MIN1,MATRIX(I,J))
      MIN2=MIN0(MIN2,NATRIX(I,J))
      MAX=MAX0(MAX,JATRIX(I,J))
      MAX1=MAX0(MAX1,MATRIX(I,J))
      MAX2=MAX0(MAX2,NATRIX(I,J))
 20   CONTINUE
 25   CONTINUE
C***************************************
C* THE ORIGINAL MATRIX IS SCALED DOWN  *
C* BETWEEN 0 & 1 AND THEN PRINTED.     *
C* THE EXTREME ROWS & COLUMNS OF ORIG-*
C* INAL MATRIX AND THOSE ROWS & COLUMNS*
C* WHICH COULD NOT BE EVALUATED ARE    *
C* FILLED WITH RESPECTIVE MAXIMUM VALUE*
C***************************************
```

```
      A=1.0/(MAX1-MIN1)
      B=A*MIN1
      DO 35 I=2,99
      DO 34 J=2,99
      SCAMAT(I,J)=(A*MATRIX(I,J))-B
34    CONTINUE
35    CONTINUE
      DO 36 I=1,100
      SCAMAT(I,100)=1.0
      JATRIX(I,100)=MAX
      NATRIX(I,100)=MAX2
      SCAMAT(100,I)=1.0
      JATRIX(100,I)=MAX
      NATRIX(100,I)=MAX2
      SCAMAT(I,1)=1.0
      JATRIX(I,1)=MAX
      NATRIX(I,1)=MAX2
      SCAMAT(1,I)=1.0
      JATRIX(1,I)=MAX
      NATRIX(1,I)=MAX2
36    CONTINUE
      DO 801 I=2,99
      NATRIX(I,2)=MIN2
      NATRIX(2,I)=MIN2
      NATRIX(I,99)=MIN2
      NATRIX(99,I)=MIN2
801   CONTINUE
      WRITE 37,NIL
37    FORMAT("1 FOLLOWING IS THE PICTURE #",I3)
      CALL CCGRAY(BATRIX)
      ****************************************************
      * THE IMAGE ENHANCED PICTURE IS SCALED              *
      * DOWN BETWEEN 0 & 1 AND THEN PRINTED.              *
      * EDGE MATRIX IS THEN SCALED DOWN AND               *
      * PRINTED.                                          *
      ****************************************************
      A=1.0/(MAX-MIN)
      B=A*MIN
      DO 22 I=1,100
      DO 21 J=1,100
      SCAMAT(I,J)=(A*JATRIX(I,J))-B
21    CONTINUE
22    CONTINUE
      WRITE 19,NIL
19    FORMAT("1 FOLLOWING IS IMAGE-ENHANCED PICTURE #",I3)
      CALL CCGRAY(BATRIX)
      A=1.0/(MAX2-MIN2)
      B=A*MIN2
      DO 18 I=1,100
      DO 17 J=1,100
      SCAMAT(I,J)=(A*NATRIX(I,J))-B
17    CONTINUE
```

```
 18     CONTINUE
        WRITE 16,NIL
 16     FORMAT("1 FOLLOWING IS THE EDGE MATRIX #",I3)
        CALL CCGRAY(BATRIX)
        ************************************************************
        * INT= THRESHOLD FOR IMAGE ENHANCED PICTURE                *
        * IEDGE= THRESHOLD FOR EDGE MATRIX                         *
        * THE SCALED DOWN MATRIX CONTAINS THE EDGE MATRIX          *
        * SO ALL THOSE PIXEL VALUES IN SCAMAT ARE MADE 0           *
        * FOR WHICH THE CORRESPONDING PIXELS IN EDGE AND           *
        *IMAGE ENHANCED MATRICES ARE NOT ABOVE THEIR RESP-         *
        *ECTIVE THRESHOLDS. THRESHOLDED PICTURE IS PRINTED         *
        ************************************************************
        INT=DEAN1+VAR1
        IEDGE=DEAN2+VAR2
        DO 39 I=2,99
        DO 38 J=2,99
        IF(JATRIX(I,J).GT.INT.AND.NATRIX(I,J).GT.IEDGE) GOTO 38
        SCAMAT(I,J)=0.0
 38     CONTINUE
 39     CONTINUE
        WRITE 13,NIL
 13     FORMAT("1 FOLLOWING IS THE THRESHOLDED PICTURE #",I3)
        CALL CCGRAY(BATRIX)
        ******************************************
        *       CONNECTEDNESS      TEST          *
        *       -------------      ----          *
        ******************************************
 830    ICOUNT=0
        DO 820 I=3,98
        DO 810 J=3,98
        NABOR=0
        IF(SCAMAT(I,J).EQ.0.0) GOTO 810
        IF(SCAMAT(I-1,J).EQ.0.0) GOTO 821
        NABOR=NABOR+1
 821    IF(SCAMAT(I,J-1).EQ.0.0) GOTO 822
        NABOR=NABOR+1
 822    IF(SCAMAT(I,J+1).EQ.0.0) GOTO 823
        NABOR=NABOR+1
 823    IF(SCAMAT(I+1,J).EQ.0.0) GOTO 824
        NABOR=NABOR+1
 824    IF(NABOR.LE.1) ICOUNT=ICOUNT+1
        IF(NABOR.LE.1) SCAMAT(I,J)=0.0
 810    CONTINUE
 820    CONTINUE
        IF(ICOUNT.NE.0) GOTO 830
        WRITE 12,NIL
 12     FORMAT("1 FOLLOWING IS THE PICTURE #",I3,"AFTER CONNECTEDNESS TEST
       +")
        CALL CCGRAY(BATRIX)
```

```
******************************************
*              SEGMENTATION              *
*              ------------              *
******************************************
      DO 449 I=1,100
      DO 445 J=1,100
      MATRIX(I,J)=0
445   CONTINUE
449   CONTINUE
      NT=0
      DO 899 I=3,98
      DO 898 J=3,98
      IF(SCAMAT(I,J).EQ.0.0.OR.MATRIX(I,J).NE.0) GOTO 898
      IF(MATRIX(I-1,J).NE.0) GOTO 897
      IF(MATRIX(I,J-1).NE.0) GOTO 896
      NT=NT+1
      MATRIX(I,J)=NT
      GOTO 898
896   MATRIX(I,J)=MATRIX(I,J-1)
      GOTO 898
897   MATRIX(I,J)=MATRIX(I-1,J)
898   CONTINUE
899   CONTINUE
891   NH=0
      DO 895 I=1,100
      DO 894 J=1,100
      IF(MATRIX(I,J).EQ.0) GOTO 894
      IF(MATRIX(I,J+1).EQ.0.OR.MATRIX(I,J+1).EQ.MATRIX(I,J)) GOTO 893
      MATRIX(I,J)=MIN0(MATRIX(I,J),MATRIX(I,J+1))
      MATRIX(I,J+1)=MATRIX(I,J)
      NH=NH+1
893   IF(MATRIX(I,J-1).EQ.0.OR.MATRIX(I,J-1).EQ.MATRIX(I,J)) GOTO 854
      MATRIX(I,J)=MIN0(MATRIX(I,J),MATRIX(I,J-1))
      MATRIX(I,J-1)=MATRIX(I,J)
      NH=NH+1
854   IF(MATRIX(I+1,J).EQ.0.OR.MATRIX(1+1,J).EQ.MATRIX(I,J)) GOTO 894
      MATRIX(I,J)=MIN0(MATRIX(I,J),MATRIX(I+1,J))
      MATRIX(I+1,J)=MATRIX(I,J)
      NH=NH+1
894   CONTINUE
895   CONTINUE
      IF(NH.NE.0) GOTO 891
      NT=0
      DO 910 I=1,100
      DO 889 J=1,100
      NT=MAX0(NT,MATRIX(I,J))
889   CONTINUE
910   CONTINUE
```

```
      ****************************************
      *              THINNING                *
      *              --------                *
      ****************************************
      DO 888 K=1,NT
      MINI(K)=100
      MINJ(K)=100
      MAXI(K)=0
      MAXJ(K)=0
      DO 887 I=1,100
      DO 886 J=1,100
      IF(MATRIX(I,J).NE.K) GOTO 886
      MINI(K)=MIN0(I,MINI(K))
      MINJ(K)=MIN0(J,MINJ(K))
      MAXI(K)=MAX0(I,MAXI(K))
      MAXJ(K)=MAX0(J,MAXJ(K))
886   CONTINUE
887   CONTINUE
888   CONTINUE
      DO 879 K=1,NT
      IF(MINI(K).EQ.100.OR.MINJ(K).EQ.100.OR.MAXI(K).EQ.0.OR.MAXJ(K).
     +   EQ.0) GOTO 879
878   N1=0
      N2=0
      N3=MINJ(K)
      N4=MAXJ(K)
      DO 877 I=N3,N4
      IF(MATRIX(MINI(K),I).EQ.0) GOTO 876
      N1=N1+1
876   IF(MATRIX(MINI(K)+1,I).EQ.0) GOTO 877
      N2=N2+1
877   CONTINUE
      IF(N1.GE.N2) GOTO 874
      DO 875 I=N3,N4
      MATRIX(MINI(K),I)=0
875   CONTINUE
      MINI(K)=MINI(K)+1
      GOTO 878
874   N1=0
      N2=0
      DO 873 I=N3,N4
      IF(MATRIX(MAXI(K),I).EQ.0) GOTO 872
      N1=N1+1
872   IF(MATRIX(MAXI(K)-1,I).EQ.0) GOTO 873
      N2=N2+1
873   CONTINUE
      IF(N1.GE.N2) GOTO 870
      DO 871 I=N3,N4
      MATRIX(MAXI(K),I)=0
871   CONTINUE
      MAXI(K)=MAXI(K)-1
      GOTO 874
870   N1=
      N2=
      N3=MINI(K)
      N4=MAXI(K)
```

```
      DO 869 I=N3,N4
      IF(MATRIX(I,MINJ(K)).EQ.0) GOTO 868
      N1=N1+1
868   IF(MATRIX(I,MINJ(K)+1).EQ.0) GOTO 869
      N2=N2+1
869   CONTINUE
      IF(N1.GE.N2) GOTO 866
      DO 867 I=N3,N4
      MATRIX(I,MINJ(K))=0
867   CONTINUE
      MINJ(K)=MINJ(K)+1
      GOTO 870
866   N1=0
      N2=0
      DO 865 I=N3,N4
      IF(MATRIX(I,MAXJ(K)).EQ.0) GOTO 864
      N1=N1+1
864   IF(MATRIX(I,MAXJ(K)-1).EQ.0) GOTO 865
      N2=N2+1
865   CONTINUE
      IF(N1.GE.N2) GOTO 862
      DO 863 I=N3,N4
      MATRIX(I,MAXJ(K))=0
863   CONTINUE
      MAXJ(K)=MAXJ(K)-1
      GOTO 866
862   N1=MINI(K)
      N2=MAXI(K)
      N3=MINJ(K)
      N4=MAXJ(K)
      NP(K)=0
      DO 861 I=N1,N2
      DO 860 J=N3,N4
      IF(I.EQ.N1.OR.I.EQ.N2.OR.J.EQ.N3.OR.J.EQ.N4) GOTO 857
      GOTO 860
857   IF(MATRIX(I,J).NE.K) GOTO 860
      NP(K)=NP(K)+1
860   CONTINUE
861   CONTINUE
      *******************************************
      *               SIZE TEST                 *
      *               ---- ----                 *
      *******************************************
      ITEST(K)=1
      LENGTH=(N2-N1)+1
      IWIDTH=(N4-N3)+1
      IAREA=((2*LENGTH)+(2*IWIDTH))*.5
      RATIO=FLOAT(LENGTH)/FLOAT(IWIDTH)
      IF(RATIO.GT.8.0.OR.RATIO.LT.0.125) ITEST(K)=0
      IF(LENGTH.LE.2.OR.IWIDTH.LE.2) ITEST(K)=0
      IF(NP(K).LT.IAREA)ITEST(K)=0
      DO 856 I=N1,N2
      DO 855 J=N3,N4
      IF(ITEST(K).EQ.0) MATRIX(I,J)=0
      IF(ITEST(K).EQ.1) MATRIX(I,J)=K
855   CONTINUE
856   CONTINUE
879   CONTINUE
```

```
      ***********************************************
      * EVERY TARGET IN THE FINAL PROCESSED PIC-*
      * TURE IS GIVEN A SEPARATE LETTER IDENTIF-*
      * ICATION AND THE PICTURE IS PRINTED.     *
      ***********************************************
      DO 848 I=1,100
      DO 847 J=1,100
      IF(MATRIX(I,J).EQ.0) MATRIX(I,J)=1H
      IF(MATRIX(I,J).EQ.1) MATRIX(I,J)=1HA
      IF(MATRIX(I,J).EQ.2) MATRIX(I,J)=1HB
      IF(MATRIX(I,J).EQ.3) MATRIX(I,J)=1HC
      IF(MATRIX(I,J).EQ.4) MATRIX(I,J)=1HD
      IF(MATRIX(I,J).EQ.5) MATRIX(I,J)=1HE
      IF(MATRIX(I,J).EQ.6) MATRIX(I,J)=1HF
      IF(MATRIX(I,J).EQ.7) MATRIX(I,J)=1HG
      IF(MATRIX(I,J).EQ.8) MATRIX(I,J)=1HH
      IF(MATRIX(I,J).EQ.9) MATRIX(I,J)=1HI
      IF(MATRIX(I,J).EQ.10) MATRIX(I,J)=1HJ
      IF(MATRIX(I,J).EQ.11) MATRIX(I,J)=1HK
      IF(MATRIX(I,J).EQ.12) MATRIX(I,J)=1HL
      IF(MATRIX(I,J).EQ.13) MATRIX(I,J)=1HM
      IF(MATRIX(I,J).EQ.14) MATRIX(I,J)=1HN
      IF(MATRIX(I,J).EQ.15) MATRIX(I,J)=1HO
      IF(MATRIX(I,J).EQ.15) MATRIX(I,J)=1HP
      IF(MATRIX(I,J).EQ.17) MATRIX(I,J)=1HQ
      IF(MATRIX(I,J).EQ.18) MATRIX(I,J)=1HR
      IF(MATRIX(I,J).EQ.19) MATRIX(I,J)=1HS
      IF(MATRIX(I,J).EQ.20) MATRIX(I,J)=1HT
      IF(MATRIX(I,J).EQ.21) MATRIX(I,J)=1HU
      IF(MATRIX(I,J).EQ.22) MATRIX(I,J)=1HV
      IF(MATRIX(I,J).EQ.23) MATRIX(I,J)=1HW
      IF(MATRIX(I,J).EQ.24) MATRIX(I,J)=1HX
      IF(MATRIX(I,J).EQ.25) MATRIX(I,J)=1HY
847   CONTINUE
848   CONTINUE
      DO 849 I=1,100
      MATRIX(I,1)=1HZ
      MATRIX(I,100)=1HZ
      MATRIX(1,I)=1HZ
      MATRIX(100,I)=1HZ
849   CONTINUE
      WRITE 2,NIL
      FORMAT("1 FOLLOWING IS THE PROCESSED PICTURE #",I3)
         DO 859 I=1,100
      WRITE 858,(MATRIX(I,J),J=1,100),I
859   CONTINUE
858   FORMAT(4X,100A1,1X,I3)
      STOP
      END
```

```
      SUBROUTINE THRESH(MATRIX,DEAN,VAR)
      DIMENSION MATRIX(100,100)
      DEAN=0
      VAR=0
      DO 890 I=3,98
      DO 790 J=3,98
      DEAN=DEAN+MATRIX(I,J)
 690  CONTINUE
 790  CONTINUE
      DEAN=DEAN/(96**2)
      DO 780 I=3,98
      DO 770 J=3,98
      VAR=VAR+((MATRIX(I,J)-DEAN)**2)
 770  CONTINUE
 780  CONTINUE
      VAR=SQRT(VAR/(96**2))
      RETURN
      END
```

```
      SUBROUTINE CCGRAY(BATRIX)
      COMMON SCAMAT(100,100)
      DIMENSION BATRIX(100,100)
      DO 500 I=1,100
      DO 450 J=1,100
      IF(SCAMAT(I,J).GT.0.175) GO TO 401
      BATRIX(I,J)=1H
      GO TO 450
401    IF(SCAMAT(I,J).GT.0.185) GO TO 402
      BATRIX(I,J)=1H-
      GO TO 450
402   IF(SCAMAT(I,J).GT.0.235) GOTO 403
      BATRIX(I,J)=1H=
      GOTO 450
403    IF(SCAMAT(I,J).GT.0.27) GO TO 404
       BATRIX(I,J)=1H+
      GOTO 450
404     IF(SCAMAT(I,J).GT.0.31) GOTO 405
      BATRIX(I,J)=1H)
      GOTO 450
405    IF(SCAMAT(I,J).GT.0.35) GOTO 406
      BATRIX(I,J)=1HI
      GOTO 450
406    IF(SCAMAT(I,J).GT.0.385) GOTO 407
      BATRIX(I,J)=1HZ
      GOTO 450
407    IF(SCAMAT(I,J).GT.0.41) GOTO 408
      BATRIX(I,J)=1HX
      GOTO 450
408    IF(SCAMAT(I,J).GT.0.435) GOTO 409
      BATRIX(I,J)=1HA
       GO TO 450
409    IF(SCAMAT(I,J).GT.0.49) GOTO 410
        BATRIX(I,J)=1HM
       GO TO 450
410    BATRIX(I,J)=1HO
450     CONTINUE
      WRITE 900,(BATRIX(I,J),J=1,100),I
900   FORMAT(1H ,100A1,10X,I2)
       DO 420 K=1,100
       IF(SCAMAT(I,K).GT.0.49) GOTO 411
       BATRIX(I,K)=1H
       GOTO 420
411    IF(SCAMAT(I,K).GT.0.545) GOTO 412
       BATRIX(I,K)=1H-
       GOTO 420
412    IF(SCAMAT(I,K).GT.0.59) GOTO 413
        BATRIX(I,K)=1H=
       GOTO 420
413    IF(SCAMAT(I,K).GT.0.82) GOTO 414
        BATRIX(I,K)=1H+
       GOTO 420
414    BATRIX(I,K)=1HX
420     CONTINUE
        WRITE 910,(BATRIX(I,K),K=1,100)
910    FORMAT(1H+,100A1)
```

```
      DO 430 L=1,100
      IF(SCAMAT(I,L).GT.0.62) GOTO 421
      BATRIX(I,L)=1H
      GO TO 430
421   BATRIX(I,L)=1H'
430    CONTINUE
       WRITE 910,(BATRIX(I,L),L=1,100)
     DO 440 M=1,100
      IF(SCAMAT(I,M).GT.0.655) GOTO 431
     BATRIX(I,M)=1H
      GO TO 440
431   BATRIX(I,M)=1H.
440    CONTINUE
       WRITE 910,(BATRIX(I,M),M=1,100)
       DO 460 N=1,100
       IF(SCAMAT(I,N).GT.0.58) GOTO 441
       BATRIX(I,N)=1H
       GOTO 460
441   IF(SCAMAT(I,N).GT.0.82) GOTO 442
      BATRIX(I,N)=1H=
       GOTO 460
442   IF(SCAMAT(I,N).GT.0.87) GOTO 443
      BATRIX(I,N)=1H-
        GOTO 460
443   BATRIX(I,N)=1HH
460    CONTINUE
      WRITE910,(BATRIX(I,N),N=1,100)
      DO 470 IN=1,100
      IF (SCAMAT(I,IN).GT.0.87) GOTO 451
      BATRIX(I,IN)=1H
     GOTO 470
451   IF(SCAMAT(I,IN).GT.0.91) GOTO 452
      BATRIX(I,IN)=1HC
     GOTO 470
452    BATRIX(I,IN)=1H3
470    CONTINUE
      WRITE 910,(BATRIX(I,IN),IN=1,100)
      DO 480 NN=1,100
      IF(SCAMAT(I,NN).GT.0.95) GOTO 461
      BATRIX(I,NN)=1H
       GOTO 480
461   BATRIX(I,NN)=1HV
480   CONTINUE
      WRITE 910,(BATRIX(I,NN),NN=1,100)
       DO 490 MN=1,100
       IF(SCAMAT(I,MN).GT.0.985) GOTO 471
       BATRIX(I,MN)=1H
      GOTO 490
471   BATRIX(I,MN)=1HA
490   CONTINUE
      WRITE 910,(BATRIX(I,MN),MN=1,100)
500   CONTINUE
     RETURN
```

Vita

Syed Naser Ali Hamadani was born in Jhang, Pakistan in 1955. After doing his matriculation and intermediate from Cadet College Kohat, he joined the Pakistan Air Force in 1973. He did his undergraduate in avionics engineering from the Pakistan Air Force College of Aeronautical Engineering. Upon completion of his engineering studies in 1977, he was commissioned in the Air Force as a Flying Officer. After serving at various positions in the Pakistan Air Force Headquarters and promotion to Flight Lieutenant, he was assigned to the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, USA.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GEO/EE/81D-3	2. GOVT ACCESSION NO. AD-A115	3. RECIPIENT'S CATALOG NUMBER 480
4. TITLE (and Subtitle) AUTOMATIC TARGET CUEING IN IR IMAGERY		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Naser A. Hamadani Flt Lt PAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AF Institute of Technology (AFIT/EN) Wright-Patterson AFB OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE December 1981
		13. NUMBER OF PAGES 71
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

15 APR 1982

18. SUPPLEMENTARY NOTES Approved for Public Release IAW AFR 190-17

Dean for Research and Professional Development

Lynch S. Wolan

FREDERICK C. LYNCH, ~~Major~~, USAF

~~Director of Public Affairs~~

Air Force Institute of Technology (ATC)

Wright-Patterson AFB, OH 45433

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Pattern Recognition — Target Cueing

Target Detection — Image Processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis presents an algorithm for detecting man-made objects embedded in low resolution imagery. A modified Kirsch edge operator is used for initial image enhancing. A normal Kirsch operator is then used for edge detection. A two-dimensional threshold for edges and the original intensity detects the pixels on the edges of the objects only. These pixels are then subjected to connectedness and size tests to detect the blobs which most

(Continued on Reverse)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

BLOCK 20: Abstract (Cont'd)

probably represent man-made objects. The algorithm was tried on 325 pictures and a detection probability of 83.3% was achieved. False alarm probability was less than 10%.

DATE

FILMED

8